Step-by-step Layered Design Generation

Faizan Farooq Khan[1], Joseph K J[2], Koustava Goswami[2], Mohamed Elhoseiny[1], Balaji Vasan Srinivasan[2]

[1]King Abdullah University of Science and Technology
[2]Adobe Research

Abstract

Design generation, in its essence, is a step-by-step process where designers progressively refine and enhance their work through careful modifications. Despite this fundamental characteristic, existing approaches mainly treat design synthesis as a single-step generation problem, significantly underestimating the inherent complexity of the creative process. To bridge this gap, we propose a novel problem setting called Step-by-Step Layered Design Generation, which tasks a machine learning model with generating a design that adheres to a sequence of instructions from a designer. Leveraging recent advancements in multi-modal LLMs, we propose SLEDGE: Step-by-step LayEred Design GEnerator to model each update to a design as an atomic, layered change over its previous state, while being grounded in the instruction. To complement our new problem setting, we introduce a new evaluation suite, including a dataset and a benchmark. Our exhaustive experimental analysis and comparison with state-of-the-art approaches tailored to our new setup demonstrate the efficacy of our approach. We hope our work will attract attention to this pragmatic and under-explored research area.

1 Introduction

Design generation is a key application at the intersection of computer vision and creative intelligence. A design is a composition of textual and visual elements harmoniously interleaved to convey intended semantics. The typical design workflow of a designer begins with a sequence of steps (see Step 1 to Step 4 in fig. 1) and uses generation tools to bring the plan to life. These tools let designers manipulate element locations and modify attributes like font size. The ability to review and update is essential in the design generation cycle.

Advances in Multi-modal Large Language Models (MLLMs) (Wu et al. 2023; Chen et al. 2023; Zhang et al. 2024b; Zhu et al. 2023; Liu et al. 2023) and diffusion-based generative models (Yang et al. 2024; Ruiz et al. 2022; Dhariwal and Nichol 2021; Ho, Jain, and Abbeel 2020) have had unprecedented success in generating graphic designs (Jia et al. 2024; Inoue et al. 2024) from textual prompts. In contrast to layout generation approaches (Lin et al. 2023a; Luo et al. 2024; Levi et al. 2023; Inoue et al. 2023b; Cheng et al. 2024) which proposes only the location of elements, design generation approaches like COLE (Jia et al. 2024), and Open-COLE (Inoue et al. 2024) proposes the content along with their layout information, making it much more closer to creating consumable designs. Two key aspects that would further improve their applicability would be: 1) the ability to consume step-by-step instructions, similar to human workflows, and 2) the ability to create layered generations, enhancing the editability of the design. These would facilitate the human-AI co-creation of graphic designs, where human creativity can be augmented with generative models.

Towards this end, we formalize and introduce the problem-setting of *Step-by-step Layered Design Generation*, inspired by how humans carry out design workflows. Given a canvas state and a natural language instruction of the intended change from a designer, the design generator should be able to generate a modified canvas aligning with the instruction. To enhance the editability of designs, each modification should be atomic and layered on top of the previous state of the canvas. Textual elements should have their attribute metadata like font size, font type, and color predicted by the model, to enhance user control.

Though intuitive, iterative design generation presents a significant challenge in practice due to its dual requirements of maintaining editability and generating cohesive content. The complexity lies not only in adding new content but also in accurately preserving various design elements, shapes, and textual components of the original canvas. As demonstrated in our experiments, existing editing models (Li et al. 2024; Feng et al. 2024) often struggle to generate content beyond localized adjustments, leading to suboptimal results in scenarios requiring holistic updates. Also, an ideal iterative design generator should be able to consume multi-modal input (canvas state and instructions) and generate multi-modal output (modified canvas state and metadata for the modification). These characteristics make the problem setup unique, and straightforward adaptation of existing approaches fails to provide acceptable fidelity as validated in section 5.

To bridge this gap, we present *SLEDGE: Step-by-step Layered Design Generator*, a novel framework by leveraging the complementary strengths of MLLMs and diffusion models to enable controlled, layerwise, iterative design generation. Our critical insight is that combining the high-level semantic understanding of MLLMs with the fine-grained

Figure 1: Motivated by how design experts create graphic designs, we introduce *a novel problem setup* and *an approach* to generate graphic designs in a step-by-step manner, seeking inputs from the user at each step. Additionally, each step generates a layer over the previous step to allow the designer to manually intervene and adjust the design if necessary. This would facilitate Human-AI co-creation, as the AI-generated content would be naively editable using design software.

generation capabilities of diffusion models enables more precise control over the design generation process. A key challenge involved in training such a composite model is curating training data. We introduce *IDeation*: Iterative Design generation dataset containing triplets of current canvas state, edit instruction, and target canvas state. IDeation Dataset lets the MLLM learn a unified representational space that can interoperate between the current state of the canvas $\mathbf{C}_t$, the modification instruction $\mathbf{I}_t$, the new state of the canvas $\mathbf{C}_{t+1}$ and the associated meta-data for modification $\mathbf{M}_{t+1}$. $\mathbf{M}_{t+1}$ corresponding to the textual changes that can be layered on top of $\mathbf{C}_t$ via deterministic text-rendering techniques, while new images added in $\mathbf{C}_{t+1}$ are extracted and harmoniously integrated into $\mathbf{C}_t$. See section 3 for details.

In order to evaluate the model on its ability to generate designs from a sequence of instructions across a diverse set of topics, we introduce *IDeation benchmark*, containing instructions across 1000 varied topics. We adapt recent state-of-the-art iterative editing approaches to the problem setup and rigorously compare against them in section 5. We find that SLEDGE is able to maintain improved fidelity and semantic consistency when compared to them.

To summarize, the key highlights of our work are:

- We introduce a novel problem: *Step-by-step Layered Design Generation*, where graphic designs would be generated and updated based on sequential user instructions.
- We propose *SLEDGE: Step-by-step Layered Design Generator*, a novel approach to address this task, leveraging multi-modal LLMs and diffusion models to achieve high-quality layered step-by-step design updates.
- We present a large-scale dataset consisting of over $150,000$ edit instructions for training and over $20,000$ instructions for testing, specifically designed to support iterative design generation.
- We introduce an evaluation benchmark with more than $1,000$ unique themes and over $10,000$ detailed edit instructions to assess the design generation performance.
- We establish a rigorous evaluation protocol to comprehensively assess the quality of generated designs across various models, both qualitatively and quantitatively.

2 Related Work

Design Generation Approaches: Layout generation methods (Chai et al. 2023; Hsu, He, and Peng 2023; Lin et al. 2023b; Shimoda et al. 2024; Yamaguchi 2021; Inoue et al. 2023a; Chen et al. 2024; Cheng et al. 2023; Guerreiro et al. 2025; Cheng et al. 2024) predict element placements either conditionally or unconditionally, producing only bounding boxes with class labels (e.g., `text`, `image`, `background`). They do not generate actual content. With MLLMs, COLE (Jia et al. 2024), and Open-COLE (Inoue et al. 2024) generate full designs from a caption in a single step, but cannot process sequential instructions due to their fixed input format. In contrast, our work is among the first to enable full-spectrum design generation driven by step-by-step designer instructions.

Iterative Generation Approaches: Diffusion models (Ho, Jain, and Abbeel 2020) excel in generative tasks and have been adapted for iterative generation and editing (Joseph et al. 2023; Feng et al. 2024; Li et al. 2024). ZONE (Li et al. 2024), extending frameworks like Instruct Pix2Pix (Brooks, Holynski, and Efros 2023; Zhang et al. 2023), supports layer-by-layer edits and serves as a key baseline for our study. Integrating LLMs with diffusion models has led to significant progress (Xia et al. 2025; Zhou et al. 2024; Feng et al. 2024; Sun et al. 2024; Ge et al. 2024). RANNI (Feng et al. 2024) uses an LLM as a planner to guide a layout-conditioned diffusion model, while LLMGA (Xia et al. 2025) refines prompts via LLMs and applies task-specific diffusion models for editing, inpainting, and T2I. We compare with both in section 5.

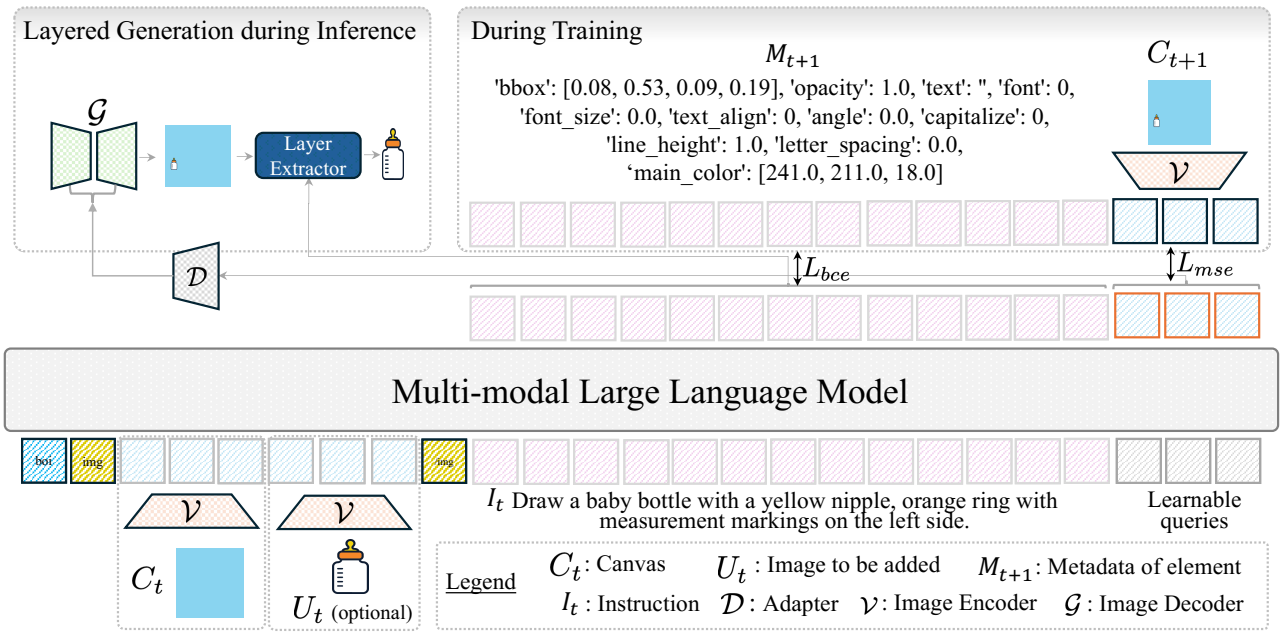

Figure 2: The figure provides an overview of SLEDGE: Step-by-step Layered Design Generator. The current state of the canvas $\mathbf{C}_t$, the instruction from the user $\mathbf{I}_t$, and an optional image to be inserted $\mathbf{U}_t$ is provided to the framework. A MLLM unifies these signals to generate the next state of the $\mathbf{C}_{t+1}$, along with the associated metadata, enabling layer-by-layer generation.

Evaluation Metrics for Generated Designs: Evaluating generated designs is challenging. Traditional metrics focus on technical aspects like layout balance (Purchase, Freeman, and Hamer 2012; Ngo, Teo, and Byrne 2003), while recent studies emphasize perceptual and aesthetic quality. FID (Heusel et al. 2018) is widely used but has known issues: sensitivity to noise (Parmar, Zhang, and Zhu 2022), diversity bias (Kynkäänniemi et al. 2023), reliance on pre-trained models (Borji 2022), and poor alignment with human perception (Chong and Forsyth 2020). To address this, we primarily adopt MLLM-based evaluation, which better captures multimodal content and aligns with human preferences (Lin et al. 2024). Haraguchi *et al.* (Haraguchi et al. 2024) further support this choice by showing strong correlation between MLLM evaluations and human judgment. Sec. 5.2 details our MLLM-based evaluation protocol. For completeness, we also report FID scores.

3 Step-by-step Layered Design Generation

We focus on the novel task of generating a graphic design by consuming step-by-step instructions from a designer. At each step of the design generation process, the current canvas state $\mathbf{C}_t$, represented as an image, a user instruction $\mathbf{I}_t$, and optionally an image to be inserted $\mathbf{U}_t$ is passed to our approach SLEDGE, denoted by $\mathcal{F}(\mathbf{C}_t, \mathbf{I}_t, \mathbf{U}_t; \boldsymbol{\theta})$ which generates the updated canvas $\mathbf{C}_{t+1}$ along with its metadata $\mathbf{M}_{t+1}$. As shown in fig. 2, the metadata contains bounding-box information of its location on the canvas. Text elements additionally include content and font information. The first canvas $\mathbf{C}_0$ is initialized blank.

Design generation necessitates a rich understanding of the interplay between the images, text, their locations, and visual aspects like color and spacing. Recent advances in multi-modal learning (Zhang et al. 2024a; Liu et al. 2023; Zhu et al. 2023; Chen et al. 2023; Sun et al. 2024), which learns a unified representation space for multi-modal entities, can provide the ideal implicit bias towards modeling designs. Hence, for instantiating $\mathcal{F}(\mathbf{C}_t, \mathbf{I}_t, \mathbf{U}_t; \boldsymbol{\theta})$, we propose to use a Multi-modal LLM architecture, explained next.

3.1 Modeling Each Canvas Generation Step

An LLM is primarily trained to process text tokens for both input and output. However, for our specific application, we require the ability to incorporate both $\mathbf{C}_t$ (canvas representations) and $\mathbf{I}_t$ (text instructions) into the LLM's workflow while also generating $\mathbf{C}_{t+1}$ as part of its outputs. This necessitates a unified representational space where visual elements (canvas) can be effectively integrated with textual information. The LLM must be able to interpret, manipulate, and condition on this shared space to generate coherent updates. We propose a three-step pipeline to achieve this:

Step 1 Aligning Visual Encoder and Decoder: We encode the visual data in our pipeline using a pre-trained Vision Transformer (Dosovitskiy et al. 2021) based image encoder, denoted by $\mathcal{V}$. The encoded visual data by $\mathcal{V}$ goes via the SD-XL decoder (Podell et al. 2023) , denoted by $\mathcal{G}(.)$. Output of $\mathcal{D}$ replaces the textual prompts passed as input to the cross-attention layers of $\mathcal{G}$ to generate back the original canvas. The parameters of both $\mathcal{V}$ and $\mathcal{G}(.)$ are frozen, while $\mathcal{D}$ is trained with reconstruction loss, ensuring a common space for the images to be projected and generated.

Step 2 Aligning Visual Encoding to MLLM's Encoding: In

this stage, we finetune the MLLM to start consuming the canvas latents from $\mathcal{V}$ along with textual data. We start off by sampling a random canvas state $\mathbf{C}_t$, the text instruction $\mathbf{I}_t$, and the image to be inserted $\mathbf{U}_t$. Note that $\mathbf{U}_t$ is optional. To simulate this during the training process, it is sampled with a probability of 0.5 and fed to $\mathcal{F}$. This mechanism enables $\mathcal{F}$ to accept additional input at inference and can act solely as a design planner. $\mathbf{C}_t$ and $\mathbf{U}_t$ are fed to $\mathcal{V}$ to get the corresponding visual embeddings before being passed onto the MLLM, as illustrated in fig. 2. The MLLM predicts both the metadata $\hat{\mathbf{M}}_{t+1}$ and the visual embeddings of the updated canvas state $\mathcal{V}(\hat{\mathbf{C}}_{t+1})$. The metadata allows for extracting the layer information as explained in section 3.2.

Our training approach treats metadata and visual embeddings differently. The metadata tokens, being textual in nature, are learned through next-token prediction with a cross-entropy loss. At the same time, visual embeddings use a Mean Squared Error (MSE) loss between the predicted embeddings and the embeddings of the ground truth updated canvas extracted from the ViT used to encode the initial canvas, shown in eq. (1). To maintain a clear separation between visual and textual embeddings, we utilize a special $\langle \texttt{img} \rangle$ token that frames start and end of visual embeddings.

$$L = L_{mse}(\mathcal{V}(\hat{\mathbf{C}}_{t+1}), \mathcal{V}(\mathbf{C}_{t+1})) + L_{bce}(\hat{\mathbf{M}}_{t+1}, \mathbf{M}_{t+1}) \tag{1}$$

Step 3 Enhancing Latent Compatibility: In this training stage, we freeze the MLLM parameters and focus on adapting the $\mathcal{D}$ module to the MLLM's output. The function of $\mathcal{D}$ is to translate the MLLM's output into a format interpretable by $\mathcal{G}(.)$, and is initially trained in Step 1. This learnable module effectively enables us to utilize $\mathcal{G}(.)$ with MLLM's multimodal output, allowing us to process and integrate the information necessary for image synthesis. We finetune $\mathcal{D}$ by freezing the MLLM and $\mathcal{G}(.)$ parameters and propagating the reconstruction loss between $\hat{C}$ and $\mathcal{C}_{\mathcal{GT}}$ only through the layers of $\mathcal{D}$. During training, $\mathcal{D}$ starts to understand the visual embeddings $\mathcal{V}(\hat{\mathbf{C}}_{t+1})$ from the MLLM while also transforming them in a manner that can be understood by $\mathcal{G}(.)$, thus acting as a bridge between MLLM and generator.

3.2 Facilitating Layered Generation

While editing a design step-by-step, only those areas that need to be modified should be altered. The rest of the design should be consistent with the previous version of the canvas.

Generating the Image Layer: To preserve the content of $\mathbf{C}_t$ while incorporating new images from $\mathbf{C}_{t+1}$, we selectively extract only the edited region from $\mathbf{C}_{t+1}$. For this, we leverage the bounding-box coordinates B predicted by the MLLM as guidance. Using these coordinates, we generate an initial binary mask $\mathbf{M}$ as defined in eq. (2), which segments the modified area from the updated canvas $\mathbf{C}_{t+1}$.

$$M(i,j) = \begin{cases} 1, & \text{if } M(i,j) \text{ is inside } B, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

However, masks from predicted bounding boxes are often coarse and may include unintended regions. To refine them, we use Segment Anything Model (SAM) (Kirillov et al. 2023) to generate high-quality masks. For each predicted mask, we compute the mean Intersection over Union(IoU) with the $\mathbf{M}$=1 region and select the one with the highest score, filtering out spurious outputs. We then dilate the selected mask to smooth the mask boundaries. The final canvas $\mathbf{C}_{t+1}$ is generated by blending the original and edited regions by:

$$\mathbf{C}_{t+1} = \mathbf{C}_t \odot (1 - \mathbf{M}) + \mathbf{C}_{t+1} \odot \mathbf{M}, \tag{3}$$

This intuitive approach ensures that only the target edit region is modified while preserving the rest of the canvas.

Generating the Text Layer: Rendering text via diffusion models is challenging, often resulting in poor legibility (Ramesh et al. 2022). Given the importance of text in design, we adopt a deterministic text rendering module (Inoue et al. 2024), bypassing generative methods. It overlays text using font and positional cues from the MLLM, ensuring clear, layout-aligned text rendering. This ensures legible text in generated images and precise alignment with the MLLM's predicted layout.

4 IDeation Dataset and Benchmark

To train and evaluate our novel problem setup, we propose *IDeation*: Iterative Design generation dataset and benchmark suite. The dataset contains $182,552$ datapoints for training, complemented by $22,881$ datapoints for testing. Each data point is a triplet containing the initial state of the canvas, edit instruction, and the corresponding modified canvas. Further, in order to capture the variety of themes in which designers would create designs, we introduce a benchmark containing $10,976$ instructions across $1,066$ design themes. We provide more details below:

4.1 IDeation Dataset

We augment the data points in the Crello dataset (Yamaguchi 2021) with turn-by-turn edit instructions to create IDeation Dataset. Crello dataset is a comprehensive resource for visual design elements sourced from the VistaCreate platform, designed to support visual synthesis tasks and layout analysis. The dataset contains over $23,000$ unique design renditions, making it one of the most extensive resources for exploring structured design composition. The dataset contains various design elements, including templates, images, icons, fonts, and layout configurations. Each design rendition is accompanied by all the individual elements associated with it. We extract and present to GPT-4o (OpenAI 2024) all the individual elements and the final composite image for each design rendition in Crello. The model is then prompted to deduce the order in which elements should be placed on an empty canvas to reconstruct the design step-by-step. Alongside deciding the order of elements, GPT-4o is asked to generate textual instructions detailing each incremental update, thereby providing a dynamic reconstruction path for the final design rendition. The exact prompt used is shown below.

> **Prompt:** You are given a list of design elements. Each element in the list is associated with a specific index

with the final element corresponding to the final design rendition. Your task is to determine the order in which these elements should be layered on a canvas in a layer-by-layer fashion. The output should be a dictionary where the key is the index of the element provided in the list, and the value is the layer to which it belongs in the final design. Additionally, for each element, provide a detailed textual instruction that can guide a text-to-image model to generate the elements on the canvas without referring to the design elements explicitly.

As a precursor to large-scale data generation, we sampled 100 design examples and manually reviewed their generated edit instructions for coherence and logical sequencing. We updated the prompts to improve the quality of the generations. Once we ensured that the generations are good for this control set, we scaled up and generated comprehensive instructions for the entire Crello (Yamaguchi 2021) dataset.

4.2 IDeation Benchmark

While the Crello (Yamaguchi 2021) dataset offers a strong base for iterative design generation, it lacks topic diversity. Its evaluation set includes only $\sim$2,000 samples across 24 categories. To address this, we propose a broader benchmark with over 1,066 unique themes and 10,000+ detailed instructions, enabling more rigorous and diverse evaluation of model generalization across varied design scenarios.

Towards creating the IDeation Benchmark, we prompt different closed-source models to generate a set of unique design themes. Specifically, we prompt Gemini (Team et al. 2023), GPT-4o (OpenAI 2024), and Claude-Sonnet (Anthropic 2024) each to give 500 different themes that can be used to create designs. We prompt different models to ensure we have a diverse set of themes. From the $1,500$ themes generated, we remove duplicates or similar themes to end-up with $1,066$ themes. They vary from "Climate Change Awareness" to "Tropical Beach Vacation".

Once we have the themes, we use in-context learning and prompt GPT-4o (OpenAI 2024) to generate edit instructions. In this way, for each theme, we have a set of instructions that we can use to generate design renditions. We evaluate the quality of edit instructions in the Ideation benchmark using GPT-4o, which confirmed 99.7% of all instructions as meaningful. The remaining instructions were removed.

5 Experiments and Results

5.1 Baseline Methods

As we introduce a novel problem setting, we adapt existing baselines to the new task and provide qualitative and quantitative comparisons with them. First, we adapt the state-of-the-art text-to-design approach Open-COLE (Inoue et al. 2024) to our setting in two ways, which we refer to as iCOLE and cCOLE. We also adapt sota iterative image editing methods to our setting. We explain them next:

iCOLE: We replace text-to-image module in Open-COLE (Inoue et al. 2024) with iterative editing module from Zone (Li et al. 2024), for step-by-step generation.

cCOLE: Here, the step-by-step instructions are concatenated, and passed together into OpenCOLE (Inoue et al. 2024). Please note that cCOLE is not iterative like other baselines. Therefore, we compare cCOLE using metrics that measure the quality of the final rendered design in table 2.

Ranni (Feng et al. 2024): combines an LLM and a diffusion model, where the LLM acts as a planner and the diffusion model as a generator. The generator is conditioned on the layout generated by the planner. It can do step-by-step editing by preserving the latents of the previous generation.

ZONE (Li et al. 2024): builds upon InstructPix2Pix (Brooks, Holynski, and Efros 2023) and MagicBrush (Zhang et al. 2023) where localization information in attention maps are combined with SAM (Kirillov et al. 2023) for layer-wise editing.

LLMGA (Xia et al. 2025): learns to refine input prompts, which is passed into diffusion models that can do inpainting, editing, and text-to-image generation. Using the image-editing part of LLMGA in a step-by-step manner for designs does not work well. This is because their editing model is optimized for localized edits and struggles when tasked with generating entirely new components. To alleviate this, we use their text-to-image model for the first step and then use the editing model. We refer to this as T2I+LLMGA. Please note that this is an *unfair advantage* to this baseline as all other baselines and our approach SLEDGE, starts off by editing the blank canvas.

5.2 Evaluation Protocol

We evaluate SLEDGE and baselines on three properties: 1) *theme adherence*, checking alignment between the final design and the theme; 2) *aesthetic quality*, evaluating the final design's aesthetics; and 3) *edit compliance*, assessing how well generations follow editing instructions. These are measured either as absolute scores on a five-point Likert scale (1=poor to 5=perfect) or as relative comparisons between SLEDGE and baselines.

Following findings in (Haraguchi et al. 2024) that MLLM evaluations align well with human judgments, we adopt state-of-the-art MLLMs GPT-4o and InternLM-XComposer-2.5 (Zhang et al. 2024a) as evaluators. For theme adherence, the MLLM receives the final design and associated design theme; for aesthetics, only the final design; and for edit compliance, the current canvas, updated canvas, and instructions.

MLLM evaluators may exhibit ordering biases (Zheng et al. 2024). To mitigate this, we use a circular evaluation strategy: each method pair is evaluated twice per output with flipped image order. This controls for bias, ensuring preferences are not order-dependent. If the MLLM's response remains unchanged, we treat the outputs as equal.

To enhance evaluation rigor, we use traditional metrics. *FID* assesses visual quality of the final design. *Aesthetic Score* is computed using a CLIP (Radford et al. 2021) model fine-tuned on LAION-Aesthetics (LAION-AI 2025). *Text Accuracy* measures CLIP similarity between predicted and reference text. *IoU* evaluates alignment between predicted and ground-truth text positions.

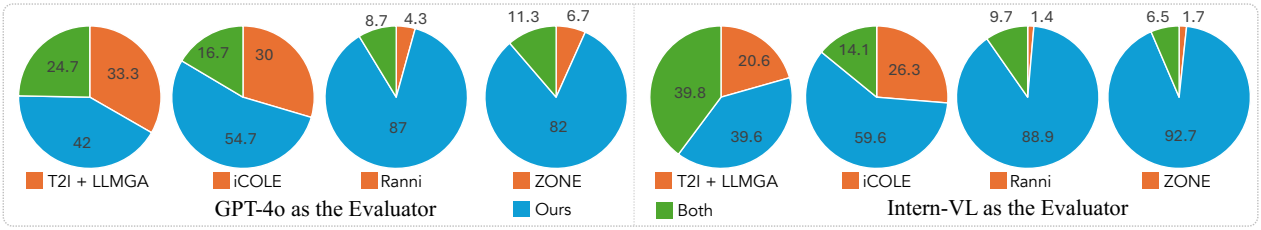

Figure 3: Performance comparison averaged across both datasets on three key aspects: theme adherence, aesthetic quality, and edit compliance. Each baseline is compared with SLEDGE, using GPT-4o and InternLM-XComposer as the evaluators.

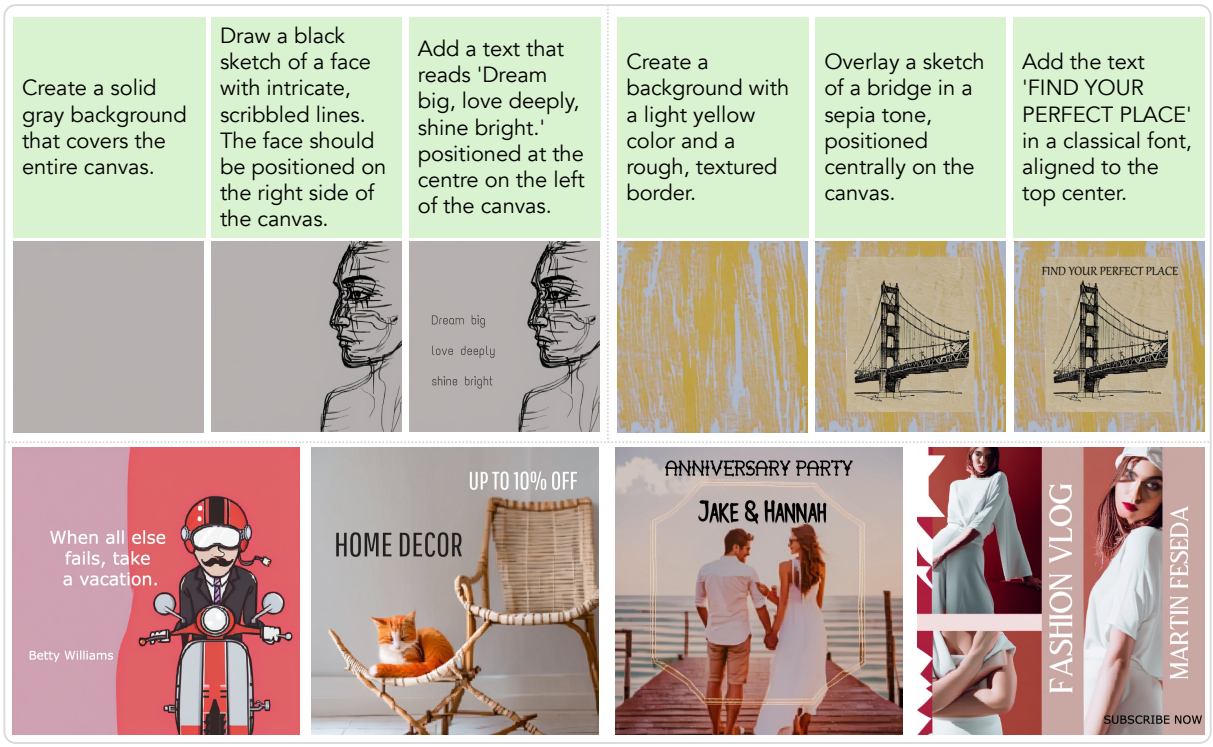

Figure 4: Illustration of iterative design generation using our method in the top row. For each design, the starting point is a blank canvas. The rest are additional final renditions from our model. Please refer to our supplementary for more examples.

5.3 Results

We showcase quantitative results in fig. 3 and tables 1 and 2 and qualitative results in figs. 4 to 6. From fig. 3 and table 1, it can be see that Ranni (Feng et al. 2024) and ZONE (Li et al. 2024) both perform poorly. We believe the main reason for the poor performance of ZONE (Li et al. 2024) could be complex edit prompts, as it is based on InstructPix2Pix (Brooks, Holynski, and Efros 2023) and MagicBrush (Zhang et al. 2023), which generally excels for simpler prompts. These complex prompts can vary from localized edits to generating new objects where ZONE does not fare well. For Ranni (Feng et al. 2024), the layout-conditioned model method offers a great solution for design generation. However, Ranni does not perform well on either of the evaluation datasets. As noted by others [1], the model size could be a problem. For qualitative results, we show results from stronger baselines(T2I+LLMGA, cCOLE, and iCOLE).

From the baselines considered, LLMGA (Xia et al. 2025) also suffers from similar problems as ZONE (Li et al. 2024). However, with T2I+LLMGA, we establish a very strong baseline. From fig. 3 and table 1, it can be seen that T2I+LLMGA and our method show competitive performance. While cCOLE is capable of generating aesthetic outputs, it lacks the layered generation capability, making it less useful for design editing operations. iCOLE performs better than ZONE (Li et al. 2024) and RANNI (Feng et al. 2024), proving to be a strong baseline, but its outputs are often not as aesthetic as SLEDGE and often too simplistic.

Further, we perform **human evaluation** using Amazon Mechanical Turk by randomly sampling 100 samples and comparing SLEDGE against baselines: T2I+LLMGA and

[1] https://github.com/ali-vilab/Ranni/issues/7

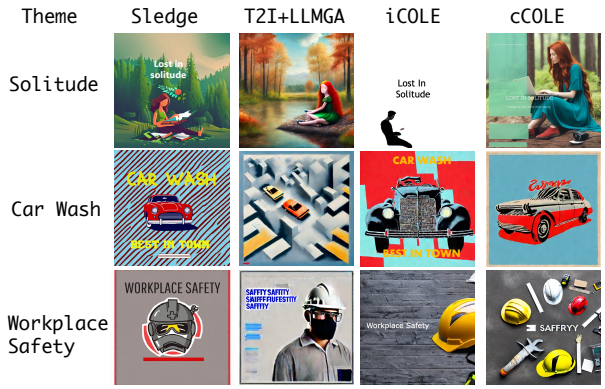

Figure 5: Qualitative comparisons show that Ranni, Zone, and cCOLE struggle, while T2I-LLMGA benefits from stronger T2I initialization; in contrast, our method achieves consistent results using a single unified model.

Figure 6: SLEDGE generates diverse samples for a given instruction by varying the random seed, using the same instructions as in fig. 4.

iCOLE. For each sample, we evaluate theme matching, aesthetic quality, and edit compliance. On average, SLEDGE was preferred by 63% of the users compared to the baselines for theme adherence. For aesthetic quality, SLEDGE was chosen by 59% of the users, and finally, 62.5% users preferred SLEDGE over the baselines for edit compliance.

5.4 Ablation Studies

We validate our design choices through an ablation study to assess the impact of key components in our approach.

Study 1 Effect of Fine-tuning $\mathcal{D}$: We compare our fine-tuned $\mathcal{D}$ module against directly using the decoder after Step 1. This allows us to evaluate the benefit of adapting $\mathcal{D}$ to our specific task. From fig. 7, we see that finetuning $\mathcal{D}$ improves the overall performance by improving alignment.

	IDeation Benchmark			Crello	
	TA(↑)	AQ(↑)	EC(↑)	AQ(↑)	EC(↑)
ZONE (Li et al. 2024)	1.23	1.37	1.84	2.22	1.84
Ranni (Feng et al. 2024)	1.50	1.51	1.23	1.72	1.72
iCOLE	2.81	3.01	2.01	3.00	2.10
T2I+LLMGA (Xia et al. 2025)	3.45	3.13	2.95	3.12	2.87
SLEDGE	**3.57**	**3.61**	**3.12**	**3.41**	**3.60**

Table 1: We compare performance across both datasets on theme adherence (TA), aesthetic quality (AQ), and edit compliance (EC). Evaluator scores range from 1 to 5.

	FID(↓)	Aesthetic Score(↑)	Text Accuracy(↑)	IoU(↑)
T2I+LLMGA	87.9	4.0	-	-
cCOLE	86.7	3.8	73.8	11.1
iCOLE	171.3	3.2	74.3	4.5
SLEDGE	**46.8**	**4.2**	**89.4**	**23.5**

Table 2: Compared with baselines SLEDGE outperforms the best baselines across FID (Heusel et al. 2018), Aesthetic Score, Text Accuracy, and IoU of textual elements.

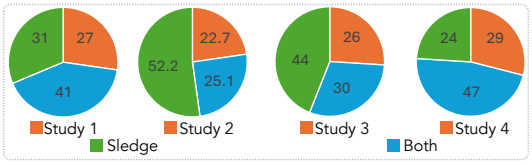

Figure 7: We compare *SLEDGE* against its ablated versions, each denoted by 'Study #', please refer to section 5.4.

Study 2 Full vs. Region Loss: We examine applying the Step 3 loss to the full predicted canvas vs. only the MLLM-specified region. The latter reduces performance, because full-image loss better leverages global context.

Study 3 $\mathcal{U}_t$ handling: We compare removing $\mathcal{U}_t$ versus replacing it with an empty canvas. Removal performs better, as empty canvas introduces misleading signals and attention overhead, while removal more clearly indicates absence.

Study 4 Layer Extractor: We examine the performance of SLEDGE without the layer extractor module. This module helps to maintain consistency across the edit instructions, ensuring that the final design is according to the user's needs. As shown in fig. 7, removal of the layer extractor significantly harms the performance.

6 Conclusion

We introduce the novel problem of *Step-by-step Layered Design Generation* and propose *SLEDGE* to address it. To support this, we set up the *IDeation evaluation suite* for robust model training and evaluation. Our experiments show that existing baselines struggle with step-by-step design prompts, while SLEDGE consistently outperforms them.

However, the diversity of the design space poses unique challenges. One is handling variable resolutions, which current generative models lack but would greatly enhance creative flexibility. Another is native support for transparency, enabling direct layer-wise generation for seamless design integration. These open directions offer promising avenues for future research, and we hope our work brings attention to this practical yet underexplored area.

Acknowledgements

This work was supported by KAUST, under Award No. BAS/1/1685-01-01.

References

Anthropic. 2024. Introducing the next generation of Claude. https://www.anthropic.com/news/claude-3-family. Accessed: 2025-02-10.

Borji, A. 2022. Pros and cons of GAN evaluation measures: New developments. *Computer Vision and Image Understanding*, 215: 103329.

Brooks, T.; Holynski, A.; and Efros, A. A. 2023. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *CVPR*.

Chai, S.; Zhuang, L.; Yan, F.; and Zhou, Z. 2023. Two-stage Content-Aware Layout Generation for Poster Designs. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, 8415–8423. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701085.

Chen, J.; Zhang, R.; Zhou, Y.; and Chen, C. 2024. Towards Aligned Layout Generation via Diffusion Model with Aesthetic Constraints. In *The Twelfth International Conference on Learning Representations*.

Chen, L.; Li, B.; Shen, S.; Yang, J.; Li, C.; Keutzer, K.; Darrell, T.; and Liu, Z. 2023. Large Language Models are Visual Reasoning Coordinators. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Cheng, C.-Y.; Huang, F.; Li, G.; and Li, Y. 2023. PLay: parametrically conditioned layout generation using latent diffusion. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Cheng, Y.; Zhang, Z.; Yang, M.; Hui, N.; Li, C.; Wu, X.; and Shao, J. 2024. Graphic Design with Large Multimodal Model. *arXiv preprint arXiv:2404.14368*.

Chong, M. J.; and Forsyth, D. 2020. Effectively Unbiased FID and Inception Score and Where to Find Them. In *CVPR*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion Models Beat GANs on Image Synthesis. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 8780–8794. Curran Associates, Inc.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929.

Feng, Y.; Gong, B.; Chen, D.; Shen, Y.; Liu, Y.; and Zhou, J. 2024. Ranni: Taming Text-to-Image Diffusion for Accurate Instruction Following. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4744–4753.

Ge, Y.; Zhao, S.; Zhu, J.; Ge, Y.; Yi, K.; Song, L.; Li, C.; Ding, X.; and Shan, Y. 2024. SEED-X: Multimodal Models with Unified Multi-granularity Comprehension and Generation. *arXiv preprint arXiv:2404.14396*.

Guerreiro, J. J. A.; Inoue, N.; Masui, K.; Otani, M.; and Nakayama, H. 2025. LayoutFlow: Flow Matching for Layout Generation. In Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T.; and Varol, G., eds., *Computer Vision – ECCV 2024*, 56–72. Cham: Springer Nature Switzerland. ISBN 978-3-031-72764-1.

Haraguchi, D.; Inoue, N.; Shimoda, W.; Mitani, H.; Uchida, S.; and Yamaguchi, K. 2024. Can GPTs Evaluate Graphic Design Based on Design Principles? In *SIGGRAPH Asia 2024 Technical Communications*, 1–4.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2018. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. arXiv:1706.08500.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. arXiv:2006.11239.

Hsu, H.; He, X.; and Peng, Y. 2023. DensityLayout: Density-Conditioned Layout GAN for Visual-Textual Presentation Designs. In *Image and Graphics: 12th International Conference, ICIG 2023, Nanjing, China, September 22–24, 2023, Proceedings, Part II*, 187–199. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-46307-5.

Inoue, N.; Kikuchi, K.; Simo-Serra, E.; Otani, M.; and Yamaguchi, K. 2023a. LayoutDM: Discrete Diffusion Model for Controllable Layout Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10167–10176.

Inoue, N.; Kikuchi, K.; Simo-Serra, E.; Otani, M.; and Yamaguchi, K. 2023b. Towards flexible multi-modal document models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14287–14296.

Inoue, N.; Masui, K.; Shimoda, W.; and Yamaguchi, K. 2024. OpenCOLE: Towards Reproducible Automatic Graphic Design Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Jia, P.; Li, C.; Yuan, Y.; Liu, Z.; Shen, Y.; Chen, B.; Chen, X.; Zheng, Y.; Chen, D.; Li, J.; Xie, X.; Zhang, S.; and Guo, B. 2024. COLE: A Hierarchical Generation Framework for Multi-Layered and Editable Graphic Design. arXiv:2311.16974.

Joseph, K. J.; Udhayanan, P.; Shukla, T.; Agarwal, A.; Karanam, S.; Goswami, K.; and Srinivasan, B. V. 2023. Iterative Multi-granular Image Editing using Diffusion Models. arXiv:2309.00613.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. *arXiv:2304.02643*.

Kynkäänniemi, T.; Karras, T.; Aittala, M.; Aila, T.; and Lehtinen, J. 2023. The Role of ImageNet Classes in Fréchet Inception Distance. In *Proc. ICLR*.

LAION-AI. 2025. LAION Aesthetic Datasets. https://github.com/LAION-AI/laion-datasets/blob/main/laion-aesthetic.md. Accessed: 2025-02-11.

Levi, E.; Brosh, E.; Mykhailych, M.; and Perez, M. 2023. Dlt: Conditioned layout generation with joint discrete-continuous diffusion layout transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2106–2115.

Li, S.; Zeng, B.; Feng, Y.; Gao, S.; Liu, X.; Liu, J.; Li, L.; Tang, X.; Hu, Y.; Liu, J.; and Zhang, B. 2024. ZONE: Zero-Shot Instruction-Guided Local Editing. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 6254–6263.

Lin, J.; Guo, J.; Sun, S.; Yang, Z.; Lou, J.-G.; and Zhang, D. 2023a. Layoutprompter: awaken the design ability of large language models. *Advances in Neural Information Processing Systems*, 36: 43852–43879.

Lin, J.; Huang, D.; Zhao, T.; Zhan, D.; and Lin, C.-Y. 2024. DesignProbe: A Graphic Design Benchmark for Multimodal Large Language Models. *arXiv preprint arXiv:2404.14801*.

Lin, J.; Zhou, M.; Ma, Y.; Gao, Y.; Fei, C.; Chen, Y.; Yu, Z.; and Ge, T. 2023b. AutoPoster: A Highly Automatic and Content-aware Design System for Advertising Poster Generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, 1250–1260. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701085.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual Instruction Tuning. arXiv:2304.08485.

Luo, C.; Shen, Y.; Zhu, Z.; Zheng, Q.; Yu, Z.; and Yao, C. 2024. Layoutllm: Layout instruction tuning with large language models for document understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15630–15640.

Ngo, D.; Teo, L.; and Byrne, J. 2003. Modelling interface aesthetics. *Information Sciences*, 152: 25–46.

OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Parmar, G.; Zhang, R.; and Zhu, J.-Y. 2022. On Aliased Resizing and Surprising Subtleties in GAN Evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11410–11420.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. arXiv:2307.01952.

Purchase, H.; Freeman, E.; and Hamer, J. 2012. An Exploration of Visual Complexity. 200–213. ISBN 978-3-642-31222-9.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv:2204.06125.

Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K. 2022. DreamBooth: Fine Tuning Text-to-image Diffusion Models for Subject-Driven Generation.

Shimoda, W.; Haraguchi, D.; Uchida, S.; and Yamaguchi, K. 2024. Towards Diverse and Consistent Typography Generation.

Sun, Q.; Cui, Y.; Zhang, X.; Zhang, F.; Yu, Q.; Wang, Y.; Rao, Y.; Liu, J.; Huang, T.; and Wang, X. 2024. Generative Multimodal Models are In-Context Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14398–14409.

Team, G.; et al. 2023. Gemini: A Family of Highly Capable Multimodal Models. Technical report, Google DeepMind.

Wu, C.; Yin, S.; Qi, W.; Wang, X.; Tang, Z.; and Duan, N. 2023. Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models. arXiv:2303.04671.

Xia, B.; Wang, S.; Tao, Y.; Wang, Y.; and Jia, J. 2025. LLMGA: Multimodal Large Language Model Based Generation Assistant. In Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T.; and Varol, G., eds., *Computer Vision – ECCV 2024*, 389–406. Cham: Springer Nature Switzerland. ISBN 978-3-031-72920-1.

Yamaguchi, K. 2021. CanvasVAE: Learning To Generate Vector Graphic Documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 5481–5489.

Yang, L.; Zhang, Z.; Yu, Z.; Liu, J.; Xu, M.; Ermon, S.; and CUI, B. 2024. Cross-Modal Contextualized Diffusion Models for Text-Guided Visual Generation and Editing. In *The Twelfth International Conference on Learning Representations*.

Zhang, K.; Mo, L.; Chen, W.; Sun, H.; and Su, Y. 2023. MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing. In *Advances in Neural Information Processing Systems*.

Zhang, P.; Dong, X.; Zang, Y.; Cao, Y.; Qian, R.; Chen, L.; Guo, Q.; Duan, H.; Wang, B.; Ouyang, L.; Zhang, S.; Zhang, W.; Li, Y.; Gao, Y.; Sun, P.; Zhang, X.; Li, W.; Li, J.; Wang, W.; Yan, H.; He, C.; Zhang, X.; Chen, K.; Dai, J.; Qiao, Y.; Lin, D.; and Wang, J. 2024a. InternLM-XComposer-2.5: A Versatile Large Vision Language Model Supporting Long-Contextual Input and Output. *arXiv preprint arXiv:2407.03320*.

Zhang, R.; Han, J.; Liu, C.; Gao, P.; Zhou, A.; Hu, X.; Yan, S.; Lu, P.; Li, H.; and Qiao, Y. 2024b. LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention. arXiv:2303.16199.

Zheng, C.; Zhou, H.; Meng, F.; Zhou, J.; and Huang, M. 2024. Large Language Models Are Not Robust Multiple Choice Selectors. In *The Twelfth International Conference on Learning Representations*.

Zhou, Y.; Zhang, R.; Gu, J.; and Sun, T. 2024. Customization Assistant for Text-to-Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9182–9191.

Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2023. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. arXiv:2304.10592.