

LayerCraft: Enhancing Text-to-Image Generation with CoT Reasoning and Layered Object Integration

Yuyao ZHANG
Dartmouth College

Jinghao LI*
CUHK

Yu-Wing TAI
Dartmouth College

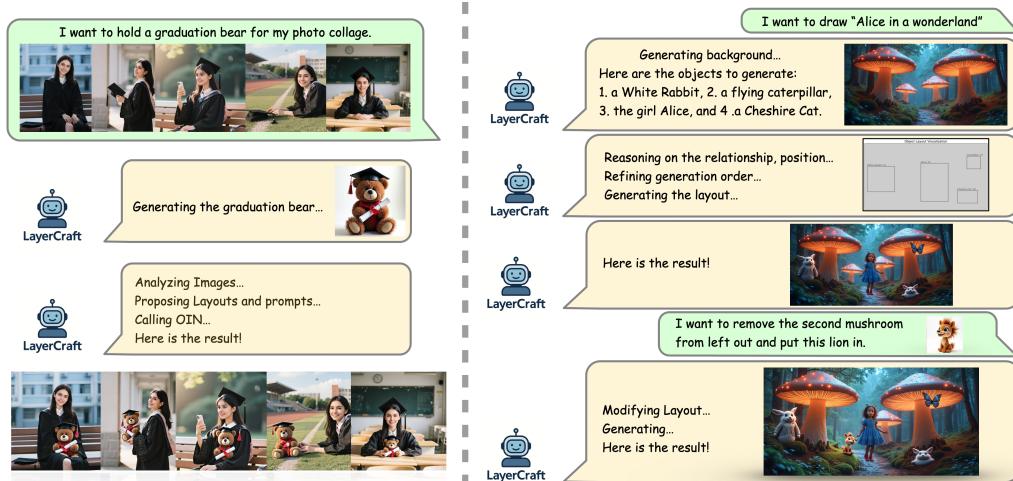


Figure 1: Application demonstrations for LayerCraft. **Left:** Demonstrates batch collage editing capabilities. A user uploads graduation photos and LayerCraft seamlessly integrates a graduation bear across all images. The system first generates a reference bear for consistency, then analyzes optimal placement while preserving facial identity and background integrity. **Right:** Illustrates the structured text-to-image generation process. From a simple "Alice in Wonderland" prompt, LayerCraft employs chain-of-thought reasoning to sequentially generate background elements, determine object layout, and compose the final image. The framework supports post-generation customization, as shown with the lion integration.

Abstract

Text-to-image (T2I) generation has made remarkable progress, yet existing systems still lack intuitive control over spatial composition, object consistency, and multi-step editing. We present **LayerCraft**, a modular framework that uses large language models (LLMs) as autonomous agents to orchestrate structured, layered image generation and editing. LayerCraft supports two key capabilities: (1) *structured generation* from simple prompts via chain-of-thought (CoT) reasoning, enabling it to decompose scenes, reason about object placement, and guide composition in a controllable, interpretable manner; and (2) *layered object integration*, allowing users to insert and customize objects—such as characters or props—across diverse images or scenes while preserving identity, context, and style. The system comprises a coordinator agent, the **ChainArchitect** for CoT-driven layout planning, and the **Object Integration Network (OIN)** for seamless image editing using off-the-shelf T2I models without retraining. Through applications like batch collage editing and narrative scene generation, LayerCraft empowers non-experts to iteratively design, customize, and refine visual content with minimal manual effort. Code will be released at <https://github.com/PeterYYZhang/LayerCraft>.

*This work was done while Jinghao Li was an exchange student at Dartmouth College.

1 Introduction

Text-to-image (T2I) generation has rapidly evolved with advances in diffusion models [14, 35, 32], transformer-based architectures [42], and scalable encoder-decoder frameworks [36]. Recent systems [9, 4, 6] produce visually impressive results from simple prompts. However, they still fall short in offering *precise, intuitive control* over spatial composition, multi-object interactions, and iterative customization.

Existing approaches to fine-grained T2I control often require architectural modifications or fine-tuning [55, 53, 37], which limits generality and usability. Others support instance-level manipulation [44, 48, 17], but often falter in complex scenes or suffer from spatial inconsistency. More structured methods like LayoutGPT [10] and GenArtist [45] attempt procedural generation, but neglect 3D spatial reasoning or rely on inefficient pipelines with excessive external tools. Even advanced multi-modal agents like GPT-4o² fail to maintain background consistency or facial identity over multiple editing iterations.

LayerCraft is our answer to these limitations: a fully automatic, modular framework for structured T2I generation and editing, designed to balance expressive control, compositional accuracy, and system efficiency. LayerCraft treats image synthesis as a step-by-step reasoning process, orchestrated by a team of specialized agents that handle prompt interpretation, spatial planning, and object integration. As shown in Figure 1, our framework supports applications such as batch collage editing with consistent object insertion, and narrative-driven image generation using structured reasoning and layout planning.

- **LayerCraft Coordinator** serves as the central interface, managing interactions between users and agents. It processes instructions, coordinates agent outputs, and integrates user feedback throughout the generation process.
- **ChainArchitect** performs *chain-of-thought (CoT)* reasoning to decompose prompts into structured layout plans. It first generates the background, then infers a spatial layout, represented as a dependency-aware 3D scene graph, to determine bounding boxes and relationships among objects. This planning phase supports complex multi-object scenes and facilitates layer-wise, editable image construction.
- **Object Integration Network (OIN)** uses the original FLUX [18] T2I model to seamlessly inpaint objects into specific regions. By applying dual LoRA adapters, OIN integrates both background and reference conditions while preserving generative quality. Its attention-mixing mechanism ensures that inserted objects align contextually and stylistically with the base image.

LayerCraft introduces several advantages over prior work: (1) it eliminates the need for model fine-tuning or external tools, making it accessible and lightweight; (2) it offers interpretable, spatially aware image construction via CoT-guided layout planning; and (3) it supports consistent object editing across single or multiple images without sacrificing visual quality. Compared to LayoutGPT [10] and GenArtist [45], which struggle with spatial coherence and integration complexity, LayerCraft provides a unified, agent-based framework capable of general-purpose generation and editing.

Our experiments demonstrate that LayerCraft excels in various creative workflows, from narrative scene composition to iterative and batch image editing, empowering both experts and non-experts to produce controllable, high-quality images with minimal effort.

2 Related Work

Controllable Image Generation. Text-to-image (T2I) generation has seen rapid progress, led by advances in diffusion models—from pixel-space methods like GLIDE [30] and Imagen [38] to more efficient latent-space frameworks such as Stable Diffusion [35] and Raphael [50]. Enhancements in multimodal alignment (e.g., DALLE-2 [33], Playground [20]) and architectural designs (e.g., Diffusion Transformers [31], PixArt [6], FLUX [18]) have substantially improved the quality and diversity of generated content. However, fine-grained and interpretable control remains challenging, especially in scenes with multiple objects or complex layouts. Personalization methods like DreamBooth [37] and Textual Inversion [11] support user-specific concepts but require task-specific

²<https://openai.com/index/gpt-4o-system-card/>

fine-tuning. Structured control approaches such as ControlNet [55] and GLIGEN [21] offer spatial conditioning via edge maps or boxes but rely on detailed inputs and lack high-level scene reasoning.

Recent techniques like Raphael [50] improve specialization through expert models at the cost of high computation. Lightweight alternatives like Attend [5] reduce overhead but struggle with compositional complexity. Autoregressive frameworks (e.g., LlamaGen [39], Show-O [49], Janus-Pro [8]) explore prompt-based synthesis via language models but often lack spatial structure. Meanwhile, emerging MLLMs such as GPT-4o and Gemini 2.0 Flash³ show generative promise, though they require substantial resources and offer limited layout control.

In contrast, LayerCraft introduces a modular, agent-based framework for structured multi-object generation and editing with minimal user input. The *ChainArchitect* employs chain-of-thought (CoT) reasoning to produce interpretable, 3D-aware layouts, enabling precise spatial planning without manual annotations. The *Object Integration Network (OIN)* leverages dual-LoRA fine-tuning on a pre-trained diffusion model (FLUX) to enable parameter-efficient object integration with strong visual fidelity and contextual coherence. Unlike methods such as OminiControl [40], which emphasize attention-based spatial aligned and subject driven generation, LayerCraft supports broader workflows, including image-guided inpainting, iterative editing, and batch collage generation, within a unified, lightweight pipeline that generalizes effectively across diverse scenarios.

Agent-Based Generation. The rise of large language models (LLMs) has greatly advanced zero- and few-shot learning across diverse domains [1, 41]. With multimodal training [2, 23, 60], LLMs have evolved into powerful agents for reasoning and creative generation [51, 47, 25, 26, 27, 16]. Among these, LayoutGPT [10] uses LLMs to generate spatial layouts from text prompts. While effective for simple scenes, its reliance on static layout models limits its ability to handle complex prompts and spatial relationships, particularly due to the absence of multi-step reasoning. Other frameworks like GenArtist [45] and LLM Blueprints [12] follow a “generate-then-edit” paradigm, refining initial layouts or images through external editing modules. This often leads to stylistic drift and unstable outputs, due to fragmented control and lack of shared context across steps.

In contrast, LayerCraft offers an integrated multi-agent framework that unifies layout planning and refinement, and iterative object integration. The *LayerCraft Coordinator* orchestrates agent interactions and incorporates user feedback throughout the process. The *ChainArchitect* improves upon LayoutGPT by applying chain-of-thought (CoT) reasoning to generate structured, 3D-aware layouts, enabling compositional planning without external layout tools. The *Object Integration Network (OIN)* complements this with image-guided inpainting via dual-LoRA fine-tuning on a pre-trained model, supporting adaptive, context-aware generation while maintaining high visual fidelity. Unlike modular pipelines that rely on third-party components or model modifications, LayerCraft remains self-contained and parameter-efficient, offering a robust and consistent user experience.

Chain of Thought Reasoning. Chain-of-thought (CoT) prompting has proven effective in improving language model reasoning by decomposing complex tasks into intermediate steps [46, 57]. However, in multimodal settings, existing CoT approaches often rely on model finetuning over specialized datasets [29, 58], limiting their applicability in zero-shot or flexible generation scenarios.

LayerCraft takes a different approach by incorporating CoT reasoning without requiring additional fine-tuning. The *LayerCraft Coordinator* uses CoT to iteratively revise and enrich user prompts, while the *ChainArchitect* applies CoT-style decomposition to translate high-level instructions into structured, 3D-aware layouts. This allows LayerCraft to reason over complex spatial relationships and multi-object configurations in a fully zero-shot, training-free setting. By leveraging CoT within a modular agent framework, LayerCraft achieves interpretable, step-wise control in multimodal image generation, offering a robust alternative to methods that depend on task-specific finetuning or static layout templates.

3 Methodology

This section elaborates the detailed design of LayerCraft, overviewed in Figure 2. Leveraging GPT-4o as the central coordinator, LayerCraft enables self-monitoring, user-agent interaction, and

³https://console.cloud.google.com/vertex-ai/publishers/google/model-garden/gemini-2.0-flash-001?inv=1&inv_t=AbxbDg

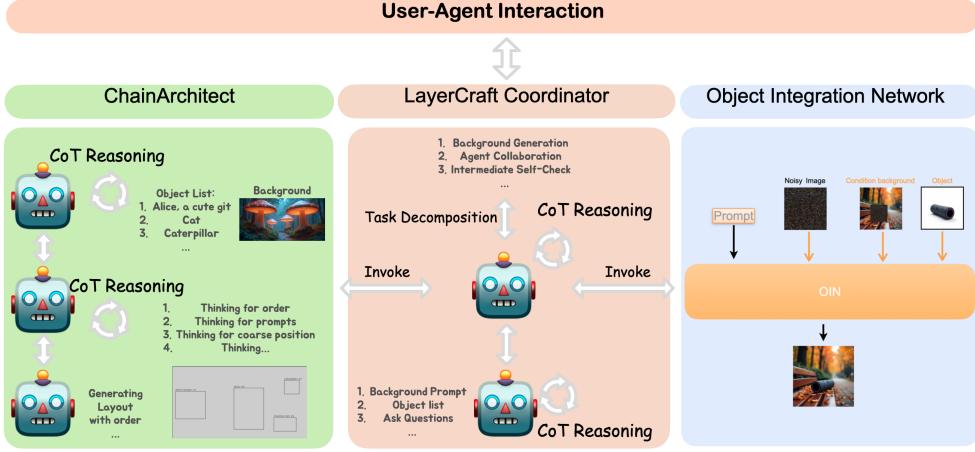


Figure 2: LayerCraft is a framework with three key components: the LayerCraft Coordinator, which processes user instructions and manages collaboration; ChainArchitect, which enhances prompts to plan layouts, identify objects and relationships, and assign bounding boxes using Chain-of-Thought reasoning; and the Object Integration Network (OIN), which enables image-guided inpainting for seamless object integration using the LoRA fine-tuned FLUX model.

aesthetically refined outputs and multi-turn editing. The framework consists of three main agents: (1) LayerCraft Coordinator, which processes user instructions and orchestrates agent collaboration; (2) ChainArchitect, a layout planning agent that generates backgrounds, assigns objects and their spatial relationships; and (3) Object Integration Network (OIN), which integrates objects seamlessly into the background based on given mask.

3.1 LayerCraft Coordinator

The LayerCraft Coordinator acts as the central orchestrator for the entire framework, overseeing the system’s operation, ensuring smooth user-agent interactions, and directing agent collaboration. This component also serves as the primary interface for user input, streamlining communication between the user and the system.

Agent-Agent Interaction The framework integrates multiple specialized agents, each responsible for a specific task such as content recognition, reference image generation, layout planning (ChainArchitect), and final image generation and inpainting (OIN). The Coordinator plays a crucial role in orchestrating these agents, breaking down tasks, assigning responsibilities, and ensuring effective communication between them. Since generative models can produce intermediate outputs with inherent randomness, the Coordinator rigorously checks the consistency of both textual and visual outputs. If discrepancies are detected, it formulates corrective measures and delegates the task to the appropriate agent for regeneration. This enables LayerCraft to ensure that the final output meets the user’s specifications.

User-Agent Interaction Although the system operates autonomously, users can modify or refine the output by interacting with the agents. For example, a user may request more details for a specific object or a customized layout for a particular region. The Coordinator facilitates multiple rounds of interaction, refining the image iteratively until the user’s requirements are fully met.

Chain-of-Thought (CoT) Enrichment. The Coordinator enhances generation by iteratively enriching the text prompt using a Chain-of-Thought (CoT) approach. Starting from the user’s input, it “asks itself” which objects should appear and how they should be arranged to meet user intent (see Figure 1, right). This reasoning produces detailed descriptions of background and foreground elements, filling gaps in the original prompt.

If the Coordinator determines the user’s prompt is already sufficiently detailed, it skips CoT reasoning and proceeds directly to task delegation. This adaptive strategy improves efficiency by avoiding unnecessary steps when the input is complete.

3.2 ChainArchitect

ChainArchitect advances traditional layout generation models (e.g., LayoutGPT [10]) by integrating Chain-of-Thought (CoT) reasoning to better handle complex prompts involving multiple objects and intricate spatial relationships.

Given a user input prompt P_i , which may range from detailed to brief, the LLM identifies relevant objects and generates a structured list $O = \{O_i \mid i \in \mathbb{N}\}$ alongside a background description P_{b_i} . For instance, if the prompt mentions a “car,” ChainArchitect infers a suitable context such as a “road.” The background description P_{b_i} is passed to the FLUX model by the Coordinator to generate the background image I_{bg} , which serves as a spatial reference for placing foreground objects.

To ensure the generated layout follows a consistent, interpretable format (e.g., JSON), ChainArchitect uses in-context exemplars 12 that define object classes, spatial positions, and scene style, thereby aligning the output with the user’s intent. Additionally, ChainArchitect leverages GPT-4o’s vision capabilities to analyze the background image viewpoint, improving object placement accuracy.

For foreground objects, ChainArchitect performs explicit spatial reasoning: it determines an optimal generation order (placing distant objects before closer ones to manage occlusion) and models inter-object relationships, such as relative positioning (“A is on top of B”) and orientation (“Person A is facing left”). This structured reasoning enables coherent and realistic multi-object layouts even in complex scenes.

3.3 Object-Integration Network (OIN)

The Object Integration Network (OIN) facilitates the seamless incorporation of objects into pre-existing backgrounds, as illustrated in Figure 3. OIN processes a masked background (delineated by a bounding box), a reference object image, and a text prompt to synthesize a contextually coherent and visually consistent integration of the specified object into the background environment.

A Parameter Reuse Method for Multiple Conditional Generation via Dual LoRA Leveraging the robust pre-trained capabilities of the FLUX text-image model on text-to-image task, we implement a parameter-efficient adaptation methodology for conditional generation. This approach enables the framework to process masked backgrounds and reference object images for highly precise subject-driven inpainting. Our implementation follows a two-phase training protocol:

In the initial phase, we develop two independent LoRA adaptors— W_{bg} and W_{obj} —that enhance the model’s capacity to interpret conditional images for inpainting and subject-driven generation tasks. Following the technique proposed in OminiControl [40], we incorporate positional embeddings for background image tokens using encodings identical to the initial noise, while reference image tokens utilize biased embeddings to accommodate spatially aligned and unaligned processing requirements.

The second phase initializes the model with the trained LoRA modules, enabling comprehensive understanding of both background and object conditions (C_{bg} and C_{obj}). To circumvent the quadratic memory complexity associated with processing extensive token sequences and to maintain clarity in condition relationships, we bifurcate the latent sequence into two components: $[C_T, X, C_{bg}]$ for background processing and $[C_T, X, C_{obj}]$ for object integration. These components undergo parallel processing with query, key, and value projections utilizing distinct weight sets: $M_{qkv}^{W_{both}}$ for joint conditions, M_{qkv} for FLUX’s foundational weights, $M_{qkv}^{W_{inp}}$ for inpainting-specialized LoRA weights,

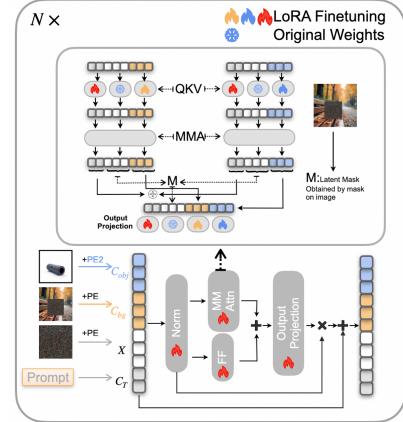


Figure 3: Architecture of the Object Integration Network (OIN). The system processes a text prompt, a background image with a designated bounding box, and a reference object to produce a seamlessly integrated result. Red, yellow, and blue indicators represent the utilization of combined LoRA weights, background inpainting weights, and subject-driven generation weights respectively. “FF” and “MM Attn” denote feedforward layers and multi-modal attention layer in the FLUX model.

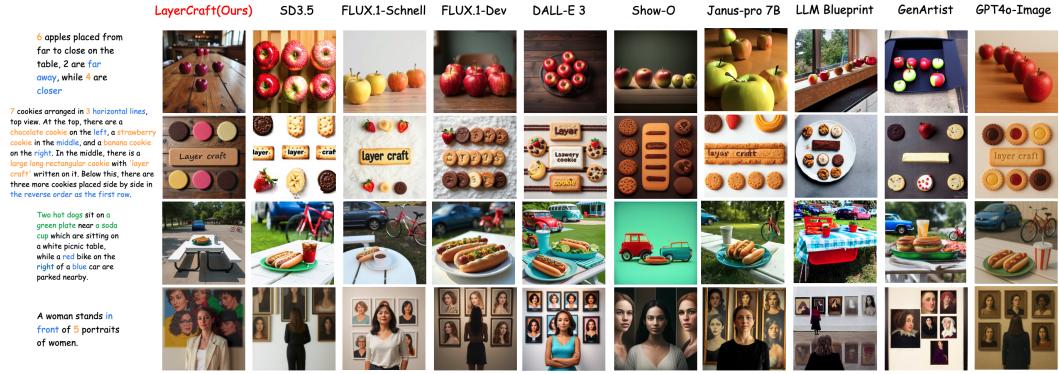


Figure 4: Visual comparisons with state-of-the-art generic text-to-image generation models are presented. On the left, the prompts are annotated with distinct colors to highlight critical attributes and relationships.

and $M_{qkv}^{W_{obj}}$ for object-specific LoRA weights. This architecture generates dual query, key, and value outputs: $[Q_1, K_1, V_1]$ for background elements and $[Q_2, K_2, V_2]$ for object features.

The attention mechanism computes outputs through the following formulations:

$$\begin{aligned} [C_T^1, X^1, C_{bg}] &= \text{Softmax} \left(\frac{Q_1 K_1^T}{\sqrt{d}} \right) V_1, \\ [C_T^2, X^2, C_{obj}] &= \text{Softmax} \left(\frac{Q_2 K_2^T}{\sqrt{d}} \right) V_2, \\ \text{Output} &= \left[\frac{C_T^1 + C_T^2}{2}, M(X^1, X^2), C_{bg}, C_{obj} \right], \end{aligned}$$

where $M(X^1, X^2)$ denotes the replacement of the masked region’s latent sequence X^1 with X^2 according to the bounding box mask. This methodology preserves both the generative capabilities of the model and its interpretation of the respective conditions, as the resultant image is generated without LoRA layer activation while conditions are processed using their corresponding LoRA weights. Consequently, the training objective focuses on establishing the relationship between textual input and the conditional elements.

Discussion: OIN supports LayerCraft’s design of the usage of intermediate reference images. It also enables LayerCraft to be an extremely efficient pipeline because during reference image creation, we employ Flux as the primary generator; when reference images are unnecessary, the LayerCraft coordinator can selectively load inpainting LoRA weights. This approach stands in contrast to frameworks such as GenArtist, which necessitates more than ten external models, resulting in computational inefficiency due to model loading/unloading cycles and introducing stylistic inconsistencies stemming from distributional variances across different models. *We will provide more details and analysis related to OIN in the supplementary materials.*

4 Experiments

Implementation Details We use OpenAI’s GPT-4o [1] as the base LLM for both the LayerCraft Coordinator and ChainArchitect agent, with the temperature set to 0.1 to balance control and creativity. Our text-to-image backbone is FLUX.1-dev [18], implemented via the Hugging Face Diffusers library [43].

The Object Integration Network (OIN) is built using Diffusers and PEFT, and trained with a batch size of 1 and gradient accumulation over 4 steps on 4 NVIDIA A6000 Ada GPUs (48GB each). We use a LoRA rank of 4 and enable gradient checkpointing for memory efficiency. OIN is trained for 20,000 iterations on a 50K subset of IPA300K, while OminiControl is fine-tuned for 50,000 iterations. Additional samples are drawn from the remaining dataset for qualitative evaluation.

Dataset Preparation (IPA300K) To ensure diversity, we use ChatGPT (via O1) to generate a list of 500 unique objects across various categories. For each object, we create 20 descriptive prompts with



Prompt: Let her wear a Chinese traditional dress.

Figure 5: More example usage of LayerCraft. We can see that our model can generate results with consistent background, and object identity comparing to GPT-4o. It also illustrates the importance of pipeline’s design with OIN and intermediate reference images. For GenArtist, even if we provide the ground truth bounding boxes and extra instructions, they still failed.

varying attributes. Following the procedure in [40], we generate 10 scene-level and 1 studio-level description per prompt to facilitate paired image generation using FLUX.1-dev with 4 random seeds. This results in paired images—one with the object in isolation and one within a complex scene. To obtain accurate object localization, we apply Grounding DINO [24] and SAM 2 [34] to extract bounding boxes from the scene images. Additional image pairs are generated with smaller object sizes to reflect realistic subject-driven inpainting cases in our framework. Bounding boxes are expanded by 15% at the bottom and 10% on each side to reduce the impact of shadows or reflections. After filtering mismatched pairs using LLM-based validation, we obtain a final dataset of 300,000 high-quality pairs, which we name **Image-guided inPainting Assets (IPA300K)**. The dataset will be released on HuggingFace.

Table 1: Comparison with other methods on T2I-Compbench [15]: The \uparrow symbol denotes that higher values correspond to better performance. Our LayerCraft system achieves the state-of-the-art performance on the benchmark.

| Method | Attribute Binding | | | Object Relationship | | Numeracy \uparrow |
|--------------------------|-------------------|------------------|--------------------|---------------------|------------------------|---------------------|
| | Color \uparrow | Shape \uparrow | Texture \uparrow | Spatial \uparrow | Non-Spatial \uparrow | |
| LayoutGPT [10] | 0.2921 | 0.3716 | 0.3310 | 0.1153 | 0.2989 | 0.4193 |
| Attn-Exct [5] | 0.6400 | 0.4517 | 0.5963 | 0.1455 | 0.3109 | - |
| GORS [15] | 0.6603 | 0.4785 | 0.6287 | 0.1815 | 0.3193 | - |
| RPG-Diffusion [52] | 0.6024 | 0.4597 | 0.5326 | 0.2115 | 0.3104 | 0.4968 |
| CompAgent [59] | 0.7400 | 0.6305 | 0.7102 | 0.3698 | 0.3104 | - |
| GenArtist [45] | 0.8482 | 0.6948 | 0.7709 | 0.5437 | 0.3346 | - |
| SDXL [32] | 0.6369 | 0.5408 | 0.5637 | 0.2032 | 0.3110 | 0.5145 |
| PixArt- α [7] | 0.6886 | 0.5582 | 0.7044 | 0.2082 | 0.3179 | 0.5001 |
| Playground v2.5 [20] | 0.6381 | 0.4790 | 0.6297 | 0.2062 | 0.3108 | 0.5329 |
| Hunyuan-DiT [22] | 0.6342 | 0.4641 | 0.5328 | 0.2337 | 0.3063 | 0.5153 |
| DALL-E 3 [3] | 0.7785 | 0.6205 | 0.7036 | 0.2865 | 0.3003 | - |
| SD v3 [9] | 0.8085 | 0.5793 | 0.7317 | 0.3144 | 0.3131 | 0.6088 |
| FLUX.1-Dev [19] | 0.7407 | 0.5718 | 0.6922 | 0.2863 | 0.3127 | 0.5872 |
| LayerCraft (Ours) | 0.8643 | 0.7046 | 0.8147 | 0.6432 | 0.3508 | 0.6331 |

4.1 Visual Comparison with State-of-the-Art Methods

Figure 4 provides a qualitative comparison of our LayerCraft framework against a diverse set of state-of-the-art baselines, including generic diffusion models [19, 3, 9], agent-based approaches [45, 12], autoregressive models [49, 8], and GPT-4o. We crafted prompts that vary in object attributes, quantities, and spatial configurations to rigorously evaluate each method’s ability to interpret and realize complex scene compositions.

Our method consistently outperforms competitors by accurately capturing both the object counts and their spatial arrangements. For example, when prompted to generate two apples positioned farther away and four apples closer to the viewpoint, LayerCraft faithfully reproduces the specified quantity and spatial layout. In contrast, while Stable Diffusion 3.5 and FLUX.1-Dev produce the correct number of apples, they fail to preserve the intended spatial relationships. GPT-4o also struggles with correct object counting, and models like PixArt- α and DALL-E 3 frequently generate incorrect object counts. Furthermore, FLUX.1-Schnell and Show-o exhibit notable errors across multiple dimensions, including color, positioning, and object consistency.

Additional visual comparisons are included in the supplementary materials. Figure 5 demonstrates LayerCraft’s effectiveness in editing collage photos via a single prompt. Compared to GPT-4o, our framework delivers superior consistency in maintaining coherent backgrounds and faithful human face details, as also illustrated in Figure 6. We further evaluate an ablation without the Object Integration Network (OIN), which forgoes intermediate reference images and results in inconsistent clothing details. Even when using a manually “hacked” version of GenArtist with ground truth bounding boxes and intermediate prompts, the output suffers from blurry faces and inconsistent attire, highlighting the critical role of our intermediate representations and integrated refinement process.

Overall, these results highlight LayerCraft’s strengths in robust multi-object control, spatial coherence, and consistent detail preservation, which collectively set it apart from prior approaches.

4.2 Comparision on T2I-Compbench

We evaluate our LayerCraft framework against two categories of state-of-the-art approaches: multi-agent systems (upper part) and generic models (lower part) on T2I-Compbench [15] since the GenEval doesn’t have the statistics for the agent based models, but we’ll include our results in the supplementary materials. As shown in Table 1, LayerCraft excels in all metrics across attribute binding, object relationship, and numeracy, outperforming others due to its instance-level control capabilities.

In contrast to agent-based generation approaches, which typically employ a “generate-then-edit” pipeline, LayerCraft generates each object sequentially under explicit positional and relational constraints. The generate-then-edit paradigm can propagate early errors into later stages, producing visible artifacts in the final output. Generic diffusion and transformer models fare even worse: lacking the ability to reason over complex textual instructions, they systematically underperform our framework across all evaluated dimensions.

4.3 More Quantitative Comparisons

To comprehensively assess the effectiveness of our approach, we conduct extensive quantitative evaluations across multiple benchmarks targeting different aspects of image generation and editing capabilities. First, we evaluate compositional generalization on the GenEval benchmark [13], which systematically tests models’ ability to handle complex compositional reasoning through six distinct subcategories (Table 2). LayerCraft achieves an overall score of 0.84, demonstrating strong performance across all evaluated dimensions. Specifically, our model achieves perfect scores (1.0) on Single Object generation, indicating robust object rendering capabilities, and excels in Two Objects composition (0.94), suggesting effective spatial reasoning. The model also shows superior color understanding with a score of 0.89 on the Colors subcategory, while maintaining competitive performance on Counting (0.82) and Position (0.75) tasks. These results notably outperform or match GPT-4o across most subcategories and significantly surpass non-agent baselines including Show-O,



Figure 6: Another example on indoor decoration, which demonstrates our model’s strong consistency.

SDXL, DALL-E 3, and SD3.5, highlighting the effectiveness of our layered integration framework for handling complex compositional requirements.

To validate real-world applicability and user preference, we conducted a comprehensive user study with 30 participants evaluating interactive complex prompt generation across diverse scenarios (Table 3). Participants assessed outputs from five state-of-the-art systems—LayerCraft, GPT-4o, LLM-Blueprint, FLUX.1 Dev, and GenArtist—using 15 carefully curated challenging prompts that span various artistic styles, compositional complexities, and semantic requirements. Each generated image was rigorously evaluated on a five-point Likert scale (1 = poor, 5 = excellent) across four critical criteria: (1) prompt consistency, measuring how accurately the generated image reflects all specified elements; (2) naturalness, assessing the realism and coherence of the generated content; (3) visual appeal, evaluating aesthetic qualities including color harmony, compositional balance, and stylistic consistency; and (4) overall quality, providing a holistic assessment. LayerCraft achieved the highest average scores in three out of four criteria, with particularly strong performance in prompt consistency (4.5/5.0) and naturalness (4.5/5.0), surpassing GPT-4o (4.4 and 4.3 respectively). LLM-Blueprint, FLUX.1 Dev, and GenArtist showed consistently lower scores across all categories, with particularly notable deficiencies in prompt consistency and visual appeal, suggesting limitations in their ability to handle complex, multi-faceted generation requirements. Additionally, to demonstrate multi-turn editing capability, we perform multi-round editing tests on the MagicBrush [54] benchmark (Table 4), where our model achieves stronger semantic alignment as reflected by higher CLIP-I/T and DINO scores. These comprehensive evaluations underscore the effectiveness of our structured multi-agent pipeline for complex generation and editing scenarios.

Table 2: Comparison of GenEval results across models. LayerCraft matches or outperforms GPT-4o on most subcategories while significantly surpassing non-agent baselines.

| Model | Overall | Single Obj. | Two Obj. | Counting | Colors | Position | Attr. Bind. |
|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| LayerCraft | 0.84 | 1.00 | 0.94 | 0.82 | 0.89 | 0.75 | 0.62 |
| GPT-4o | 0.84 | 0.99 | 0.92 | 0.85 | 0.92 | 0.75 | 0.61 |
| Show-O | 0.53 | 0.95 | 0.52 | 0.49 | 0.82 | 0.11 | 0.28 |
| SDXL | 0.55 | 0.98 | 0.74 | 0.39 | 0.85 | 0.15 | 0.23 |
| FLUX.1-Dev | 0.66 | 0.98 | 0.81 | 0.74 | 0.79 | 0.22 | 0.45 |
| DALL-E 3 | 0.67 | 0.96 | 0.87 | 0.47 | 0.83 | 0.43 | 0.45 |
| SD3.5 | 0.71 | 0.98 | 0.89 | 0.73 | 0.83 | 0.34 | 0.47 |
| Janus-pro 7B | 0.80 | 0.99 | 0.89 | 0.59 | 0.90 | 0.79 | 0.66 |

Table 3: Mean (μ) and standard deviation (σ) of user ratings (1–5). Higher values indicate better performance.

| System | Consistency | Naturalness | Visual Appeal | Overall |
|---------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| LayerCraft | 4.5 ± 0.82 | 4.5 ± 0.73 | 4.4 ± 0.78 | 4.2 ± 0.75 |
| GPT-4o | 4.4 ± 0.79 | 4.3 ± 0.86 | 4.5 ± 0.67 | 4.2 ± 0.84 |
| LLM-Blueprint | 3.0 ± 1.28 | 2.9 ± 1.09 | 2.6 ± 1.15 | 2.9 ± 0.94 |
| FLUX.1 Dev | 3.2 ± 1.38 | 3.6 ± 0.97 | 3.5 ± 1.12 | 3.3 ± 1.07 |
| GenArtist | 3.1 ± 1.08 | 3.5 ± 1.18 | 3.6 ± 1.26 | 3.5 ± 1.02 |

Table 4: Multi-turn editing results on MagicBrush. Lower is better for L1/L2; higher is better for CLIP-I, DINO, and CLIP-T.

| Method | L1 ↓ | L2 ↓ | CLIP-I ↑ | DINO ↑ | CLIP-T ↑ |
|-------------------|---------------|---------------|---------------|---------------|---------------|
| HIVE | 0.1521 | 0.0557 | 0.8004 | 0.6463 | 0.2673 |
| InstructPix2Pix | 0.1584 | 0.0598 | 0.7924 | 0.6177 | 0.2726 |
| MagicBrush | 0.0964 | 0.0353 | 0.8924 | 0.8273 | 0.2754 |
| GenArtist | 0.0858 | 0.0298 | 0.9071 | 0.8492 | 0.3067 |
| LayerCraft (ours) | 0.0863 | 0.0299 | 0.9121 | 0.8541 | 0.3157 |

5 Ablation Study

Ablation on CoT Variants To rigorously assess the contribution of Chain-of-Thought (CoT) reasoning in our layout generation process, we conducted a comprehensive ablation study by comparing

Table 5: Ablation Study for CoT on T2I-Compbench.

| Method | Attribute Binding | | | Object Relationship | | Numeracy↑ |
|-------------------------------|-------------------|---------------|---------------|---------------------|---------------|---------------|
| | Color↑ | Shape↑ | Texture↑ | Spatial↑ | Non-Spatial↑ | |
| LayerCraft | 0.8643 | 0.7046 | 0.8147 | 0.6432 | 0.3508 | 0.6331 |
| w/o generation order | 0.8524 | 0.6792 | 0.7853 | 0.4210 | 0.3147 | 0.6305 |
| w/o object relationship | 0.8512 | 0.6867 | 0.7842 | 0.4062 | 0.2854 | 0.6301 |
| w/o order & relationship | 0.8413 | 0.6463 | 0.7531 | 0.3847 | 0.2752 | 0.6023 |
| w/o CoT for Layout Generation | 0.6394 | 0.5639 | 0.7216 | 0.2831 | 0.3013 | 0.5663 |

the full LayerCraft pipeline with systematically simplified variants. The purpose is to quantify the individual and collective impact of key CoT components on generation quality and spatial coherence. Specifically, we evaluated the following variants:

- **Without Generation Order:** Removes the CoT-driven ordering mechanism used to determine the sequence of object placement.
- **Without Object Relationship:** Omits relational reasoning such as spatial prepositions or inter-object dependencies.
- **Without Both Order and Relationship:** Disables both sequential placement and object relationship modeling.
- **Without All CoT for Layout Generation:** Fully removes CoT reasoning from the ChainArchitect, falling back to a single-pass layout prediction without relationships with background.

Due to computational constraints, we employed a stratified sampling strategy and evaluated the models on 20% of the test data, ensuring balanced representation across object types and scene configurations.

As shown in Table 5, the complete LayerCraft pipeline consistently outperforms all ablated versions across key metrics, including object count accuracy, spatial arrangement fidelity, and overall realism. Notably, the absence of generation order and relationship reasoning leads to degraded spatial coherence and increased placement conflicts. The full removal of CoT results in the most significant performance drop, underscoring the critical role of iterative reasoning in managing compositional complexity.

Limitations While LayerCraft delivers strong spatial control and compositional accuracy, its use of Chain-of-Thought reasoning and multi-agent coordination introduces additional computational overhead. This can impact efficiency, particularly for complex scenes with many interacting objects. Although spatial accuracy is enhanced through background-guided bounding boxes, the primary cost lies in maintaining agent interactions. Future work will focus on streamlining these processes to improve runtime performance while preserving generation quality.

6 Conclusion

We have presented LayerCraft, a novel agent-based framework for text-to-image (T2I) generation that addresses key challenges in compositional control, spatial reasoning, and multi-object fidelity. By integrating three specialized agents, *LayerCraft Coordinator*, *ChainArchitect*, and *Object Integration Network (OIN)*, our system supports structured planning, iterative reasoning, and object-aware image refinement in a fully automated pipeline.

LayerCraft excels in generating complex scenes with accurate spatial layouts and consistent object attributes, all without requiring model finetuning. It also enables consistent multi-image editing, making it particularly effective for tasks such as photo collage editing from a single prompt. Extensive experiments demonstrate superior performance over existing methods in both accuracy and visual coherence.

With instance-level control, real-time interactivity, and a modular design, LayerCraft offers a scalable and user-friendly solution for high-quality image synthesis across a wide range of creative and practical applications.

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.
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.
- [3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn.openai.com/papers/dall-e-3.pdf*, 2(3):8, 2023.
- [4] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.
- [5] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023.
- [6] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- σ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024.
- [7] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- α : Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023.
- [8] 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.
- [9] 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.
- [10] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36:18225–18250, 2023.
- [11] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.
- [12] Hanan Gani, Shariq Farooq Bhat, Muzammal Naseer, Salman Khan, and Peter Wonka. LLM blueprint: Enabling text-to-image generation with complex and detailed prompts. In *The Twelfth International Conference on Learning Representations*, 2024.
- [13] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.
- [14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [15] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.
- [16] Alexander Huang-Menders, Xinhang Liu, Andy Xu, Yuyao Zhang, Chi-Keung Tang, and Yu-Wing Tai. Smartavatar: Text-and image-guided human avatar generation with vlm ai agents. *arXiv preprint arXiv:2506.04606*, 2025.
- [17] Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. Dense text-to-image generation with attention modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7701–7711, 2023.

[18] Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024.

[19] Black Forest Labs. Flux.1 [dev]. <https://huggingface.co/black-forest-labs/FLUX.1-dev>, 2024. A 12 billion parameter text-to-image model available under a non-commercial license.

[20] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2. 5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024.

[21] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023.

[22] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. *arXiv preprint arXiv:2405.08748*, 2024.

[23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[24] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[25] Xinhang Liu, Yu-Wing Tai, and Chi-Keung Tang. Agentic 3d scene generation with spatially contextualized vlms. *arXiv preprint arXiv:2505.20129*, 2025.

[26] Xinhang Liu, Yu-Wing Tai, and Chi-Keung Tang. Chatcam: Empowering camera control through conversational ai. *Advances in Neural Information Processing Systems*, 37:54483–54506, 2025.

[27] Xinhang Liu, Chi-Keung Tang, and Yu-Wing Tai. Worldcraft: Photo-realistic 3d world creation and customization via llm agents. *arXiv preprint arXiv:2502.15601*, 2025.

[28] Chaojie Mao, Jingfeng Zhang, Yulin Pan, Zeyinzi Jiang, Zhen Han, Yu Liu, and Jingren Zhou. Ace++: Instruction-based image creation and editing via context-aware content filling. *arXiv preprint arXiv:2501.02487*, 2025.

[29] Debjyoti Mondal, Suraj Modi, Subhadarshi Panda, Rituraj Singh, and Godawari Sudhakar Rao. Kam-cot: Knowledge augmented multimodal chain-of-thoughts reasoning. *arXiv preprint arXiv:2401.12863*, 2024.

[30] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[31] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[32] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[33] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[34] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[37] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dream-booth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

[38] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[39] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

[40] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. *arXiv preprint arXiv:2411.15098*, 3, 2024.

[41] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricu, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[42] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[43] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, Steven Liu, William Berman, Yiyi Xu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>. Diffusers provides pretrained diffusion models across multiple modalities, such as vision and audio, and serves as a modular toolbox for inference and training of diffusion models. If you use this software, please cite it using the metadata from this file.

[44] Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6232–6242, 2024.

[45] Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. *Advances in Neural Information Processing Systems*, 37:128374–128395, 2025.

[46] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[47] Qi Wu, Yubo Zhao, Yifan Wang, Xinhang Liu, Yu-Wing Tai, and Chi-Keung Tang. Motion-agent: A conversational framework for human motion generation with llms. *arXiv preprint arXiv:2405.17013*, 2024.

[48] Jinheng Xie, Yuxiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7452–7461, 2023.

[49] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[50] Zeyue Xue, Guanglu Song, Qiushan Guo, Boxiao Liu, Zhuofan Zong, Yu Liu, and Ping Luo. Raphael: Text-to-image generation via large mixture of diffusion paths. *Advances in Neural Information Processing Systems*, 36, 2024.

[51] Deshun Yang, Luhui Hu, Yu Tian, Zihao Li, Chris Kelly, Bang Yang, Cindy Yang, and Yuexian Zou. Worldgpt: a sora-inspired video ai agent as rich world models from text and image inputs. *arXiv preprint arXiv:2403.07944*, 2024.

[52] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and Bin Cui. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. In *International Conference on Machine Learning*, 2024.

[53] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

[54] Kai Zhang, Lingbo Mo, Wenhui Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.

- [55] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.
- [56] Yuxuan Zhang, Yirui Yuan, Yiren Song, Haofan Wang, and Jiaming Liu. Easycontrol: Adding efficient and flexible control for diffusion transformer. *arXiv preprint arXiv:2503.07027*, 2025.
- [57] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *International Conference on Learning Representation*, 2023.
- [58] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.
- [59] Wang Zhenyu, Xie Enze, Li Aoxue, Wang Zhongdao, Liu Xihui, and Li Zhenguo. Divide and conquer: Language models can plan and self-correct for compositional text-to-image generation. *arXiv preprint arXiv:2401.15688*, 2024.
- [60] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: **[Yes]**

Justification: The abstract and introduction clearly state the main claims. These claims are supported by experiments in Section 4.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: **[Yes]**

Justification: Section 5 explicitly discusses limitations such as additional computational overhead caused by CoT reasoning and cross-agent communication.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not present formal theorems or proofs. It includes algorithmic modifications and empirical results, not theoretical contributions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 4 describes datasets, APIs, evaluation metrics, and hardware setup. The methodology also details architecture and module integration. In our supplemental, we will include a table of all prompt texts and seeds.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: The paper builds on publicly available pretrained models. Our code, instructions and dataset will be released alongside the final submission in the supplemental.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: All training and evaluation details are provided in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[NA\]](#)

Justification: Results primarily focused on qualitative comparisons. The paper does not report error bars or statistical significance tests.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: Section 4 explicitly states all experiments used 4 NVIDIA RTX 6000 Ada GPU (48GB).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: The research presented in this paper adheres to the NeurIPS Code of Ethics. All datasets used are properly cited or created from scratch, no human subjects or sensitive personal data are involved, and the work does not pose foreseeable risks related to safety, fairness, privacy, or misuse.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: Broader societal impacts, including benefits such as accessibility and creative empowerment, as well as risks such as misuse for disinformation, will be discussed in the supplementary materials, along with recommendations for responsible use and model deployment.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: Safeguards will be implemented and detailed upon the release of the trained model. These include measures such as gated access, usage restrictions, and output watermarking to mitigate risks of misuse. The paper acknowledges potential concerns around image generation and editing, and commits to responsible deployment practices aligned with community standards.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper uses Flux and GPT-4o API, both under open-source licenses. These are cited in the references and licensed appropriately.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: **[Yes]**

Justification: The paper introduces a new agent pipeline (Layercraft), Object Integration Network (OIN) and dataset (IPA300K). We will provide code, model weights, datasets and integration instructions in the supplemental.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: **[NA]**

Justification: The paper does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Not applicable; no human subjects research is involved.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The large language models (LLMs) are a core component of this work. All LLMs usage are clearly explained in the paper.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

7 Broader Impact Statement

LayerCraft significantly advances text-to-image (T2I) generation by providing precise control over composition and object integration, driven by Chain-of-Thought (CoT) reasoning. This research promises substantial positive societal impact, primarily by **democratizing visual content creation** for non-experts, making sophisticated design tools accessible to a broader audience. It will **revolutionize creative and professional workflows** in industries like advertising and gaming, drastically accelerating content creation and fostering innovation. This capability also **catalyzes new forms of digital storytelling and education**, enabling richer visual narratives. While acknowledging risks like misuse for misinformation or bias propagation, which we condemn and will address through ethical guidelines and further research, our core focus remains on LayerCraft’s transformative power to empower human creativity and broadly benefit society.

8 More Examples on Batch Collage Editing

In this section, we present additional examples of batch collage image editing. Figure 7 illustrates LayerCraft’s ability to seamlessly integrate a Van Cleef necklace across multiple photos of a girl. Our model first intelligently identifies optimal placement bounding boxes for the necklace, then generates a consistent reference image to ensure uniformity throughout the process before engaging the Object Integration Network (OIN) for the final result. In contrast, GPT-4o struggles with this task, failing

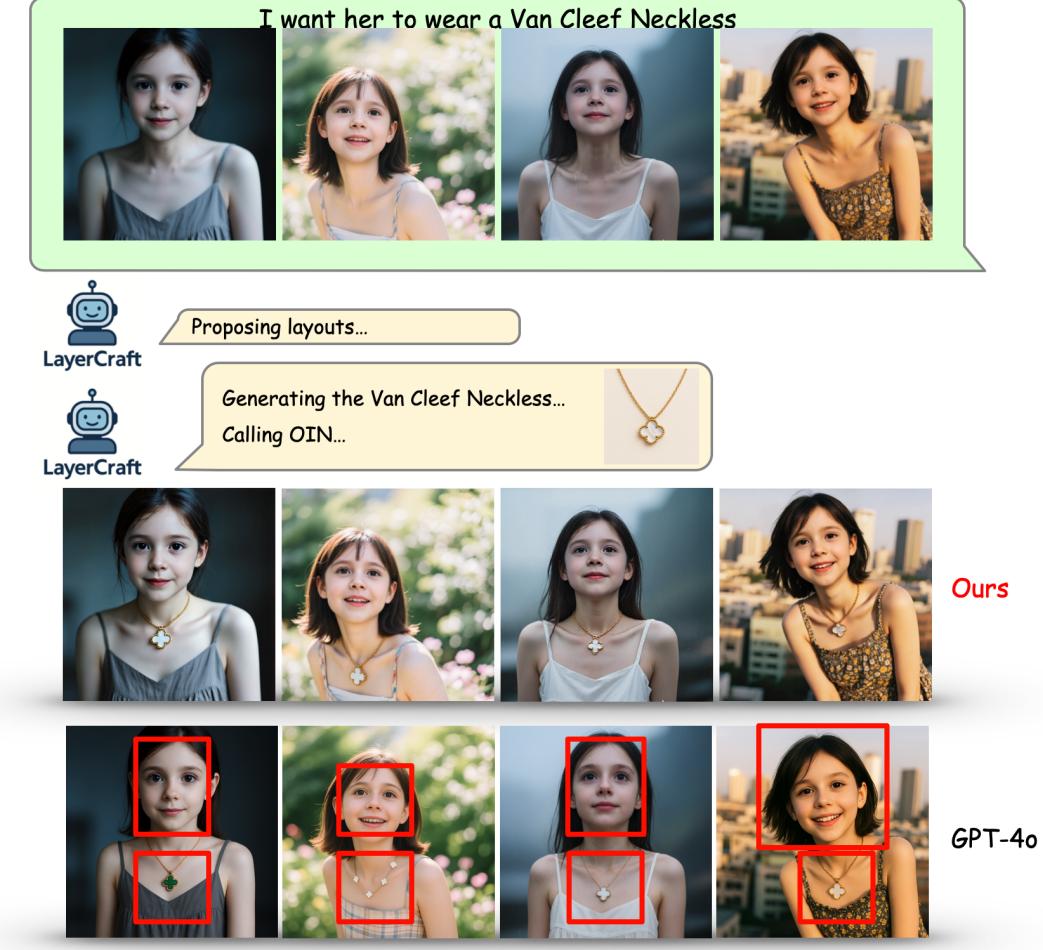


Figure 7: An example of batch collage image editing. LayerCraft effectively proposes bounding boxes for necklace placement and generates a consistent reference image, leading to seamless integration across multiple images with a single prompt. In contrast, GPT-4o fails to preserve facial identity and generates inconsistent necklaces as highlighted by the red boxes.

to preserve facial identity and generating inconsistent necklaces, as highlighted by the red boxes. Figure 8 provides further demonstrations of our model’s robust capabilities. The upper panel shows the zoomed-in version of the teaser image. The lower panel showcases a striking outfit modification, seamlessly changing a black man’s attire to a white blazer. Figure 9 effectively illustrates the generation of a cohesive Audi advertisement, featuring a single car consistently integrated across five distinct scenes.

9 Additional comparisons on T2I generation with other SOTA methods

In this section, Figure 15 presents a detailed qualitative comparison with state-of-the-art methods, including expanded versions of examples from the main paper for clearer visualization. As demonstrated, our model consistently yields better results, particularly in terms of object numeracy and accurate spatial relationships. Furthermore, LayerCraft exhibits significantly fewer artifacts compared to other agent-based methods. For instance, LLM Blueprint generates an anomalous red object beneath the table in the hot dog example. GenArtist, even in its teaser image, struggles with perspective accuracy: while the hotdogs are in focus, the distant car and bike remain sharply defined despite the blurry far end of the table, diminishing overall realism. Our method, conversely, avoids such inconsistencies, producing more coherent and realistic compositions.



Generating the graduation bear...



Analyzing Images...
Proposing Layouts and prompts...
Calling OIN...
Here is the result!



I want him to wear a white blazer



Proposing layouts...



Generating the white blazer...
Calling OIN...

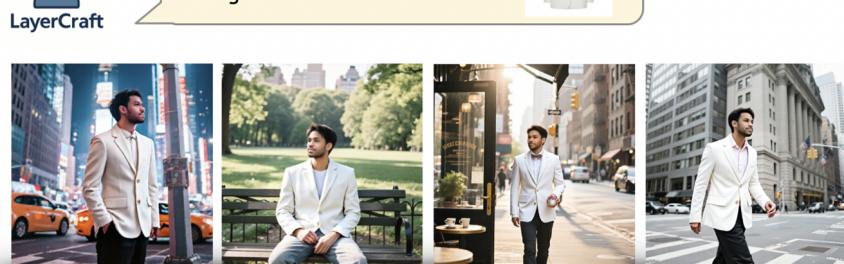


Figure 8: This figure provides more user scenario, the upper part is a larger and clearer demo for the teaser image. The lower panel demonstrates an outfit modification, showing a black man's attire seamlessly changed to a white blazer. These examples highlight the robust capabilities of our model.

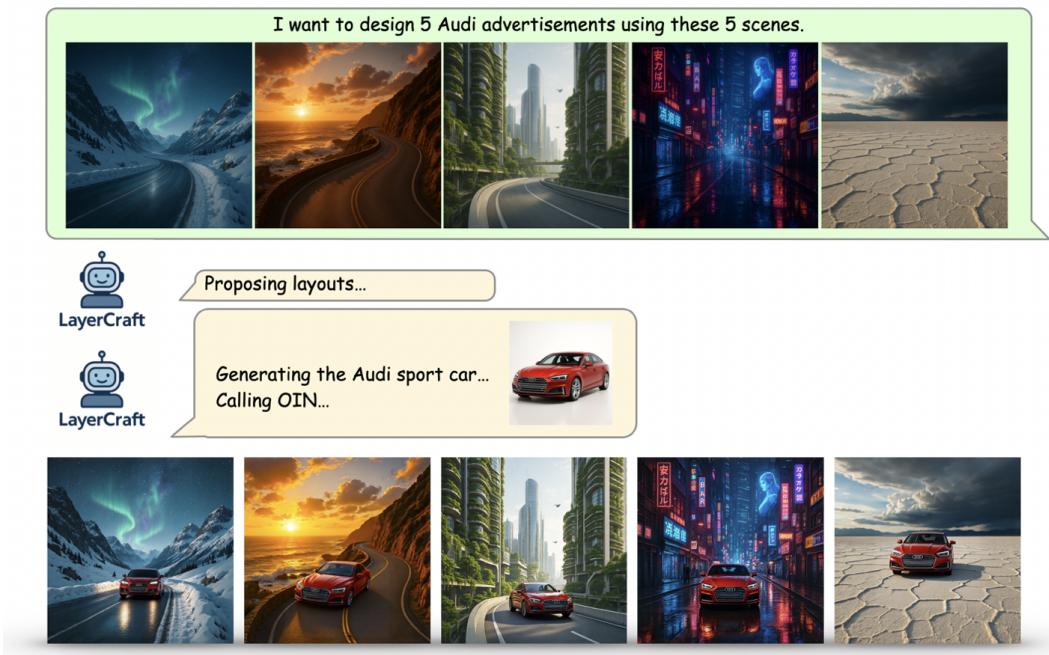


Figure 9: Further examples of batch collage image editing and generation. It illustrates the generation of a consistent Audi advertisement featuring a single car across five distinct scenes.



Figure 10: Failure case for direct attention mix in Section 10.1 “Analysis of Objection Integration Network”. One can see the background is changed and the boundary is easy to see.

10 Additional Analysis on Objection Integration Network

10.1 Ablation on Attention Mixing

In our work, attention outputs are derived through the strategic blending of dual attention maps utilizing mask indices in the latent space. Specifically, our approach computes attention outputs independently for each branch and subsequently integrates the hidden states according to a latent mask derived from the original masked region. This strategic integration ensures that the model effectively learns optimal object placement within the background while preserving background integrity. Furthermore, since the Multi-Modal attention mechanism within the FLUX architecture processes textual and image tokens concurrently, we address the cross-modal correlations by implementing a weighted average of textual tokens from both branches. This dual-branch integration synergistically enhances the model’s comprehension of the conditional inputs. To validate our approach, we conducted comparative analyses against two alternative methodologies. The first alternative, inspired by OmniControl [40], involves extending the input sequence and computing the attention matrix for the entire augmented sequence. However, this methodology encountered significant convergence

challenges during optimization. The second approach implements a weighted summation of attention outputs, which resulted in artifacts characterized by pronounced boundaries in the masked regions and unintended modifications to background elements. Figure 10 presents visual evidence of these failure cases comparing with the successful results obtained using our method. Our proposed methodology demonstrates superior performance compared to these alternatives, as illustrated in more results of OIN in Figures 13 and 14.

10.2 Comparisons with Concurrent Subject-driven Inpainting Methods

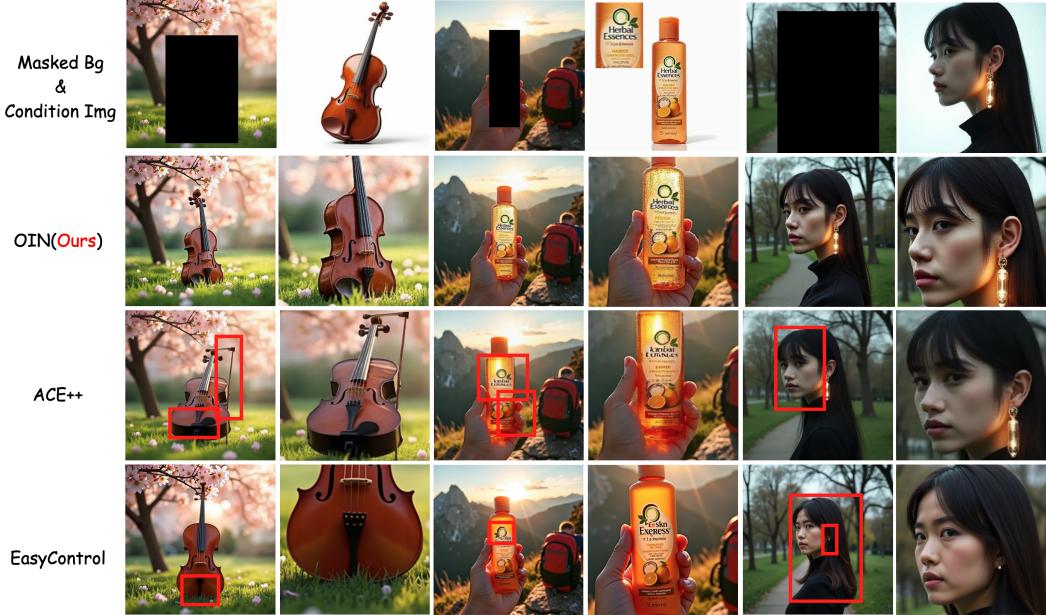


Figure 11: Visual comparisons with concurrent works on subject-driven inpainting task. The first row is our Object-Integration Network, the second row is ACE++ [28], and the last row is EasyControl [56]

In addition, we provide qualitative comparisons with concurrent approaches that support subject-driven inpainting to contextualize our contributions as being on par with, or even surpassing, the current state of the art [28, 56]. Figure 11 displays these results. A detailed examination reveals that our Object Integration Network (OIN) excels in capturing finer details, as exemplified by the orange bottle where text and logos are well-preserved, unlike the blurred text in ACE++’s output or the reverted logo in EasyControl’s result. Another instance is the violin, whose features (e.g., the chin rest) are maintained only by OIN, while they disappear in the results from ACE++ and EasyControl. Similarly, the lighted earring on the lady vanishes in EasyControl’s output. While these concurrent methods could theoretically replace OIN within our pipeline, OIN offers superior efficiency by requiring only a single loading and unloading of models, as it only necessitates loading the correct LoRA weights. In terms of speed, OIN and ACE++ generate results within one minute, depending on GPU type, whereas EasyControl is two to three times slower, further underscoring OIN’s significant contribution to the overall pipeline.

11 Handling Difficult Bounding Box Proposal.

To propose a correct bounding box using LLM only will be difficult sometimes, so we adopt a iterative refinement process to solve this problem. Figure 12 demonstrates the process. Initially, the ChainArchitect proposes an initial bounding box. This proposal is then visualized directly on the image, allowing the LLM to iteratively refine its size and position. This refinement process proves highly effective in achieving appropriate bounding box localization.

Bounding Box Refinement...



Figure 12: Our approach to handling bounding box proposals in editing tasks demonstrates remarkable effectiveness. Initially, the Large Language Model (ChainArchitect) proposes an initial bounding box. This proposal is then visualized directly on the image, allowing the LLM to iteratively refine its size and position. This refinement process proves highly effective in achieving appropriate bounding box localization.

12 Overall Process Decomposed Demo

In this section we will demonstrate how the process happen within the framework. We use the example of living room decoration to demonstrate the CoT thinking process of the framework, it is demonstrated at the end of the supplementary materials.

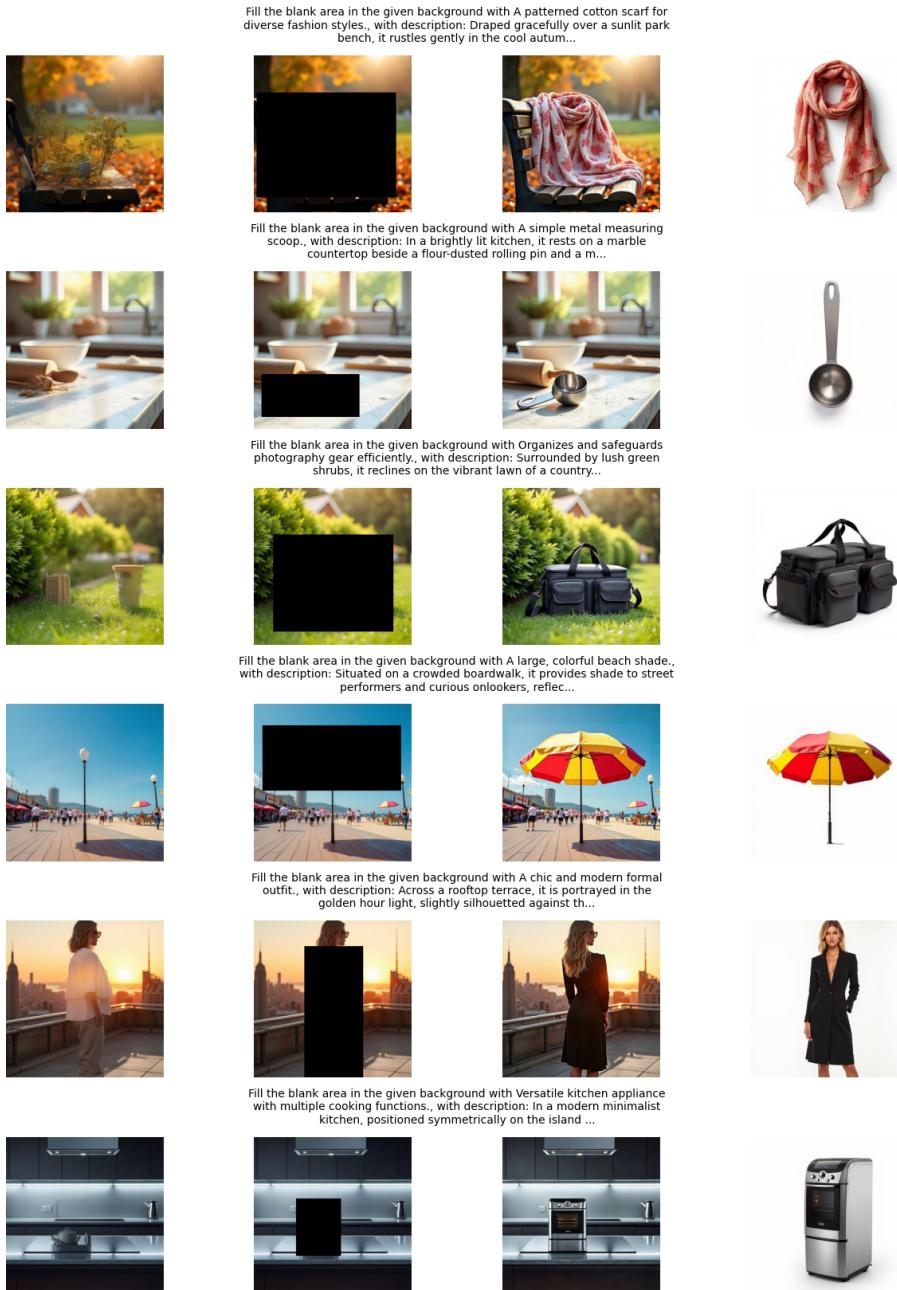


Figure 13: More results of OIN

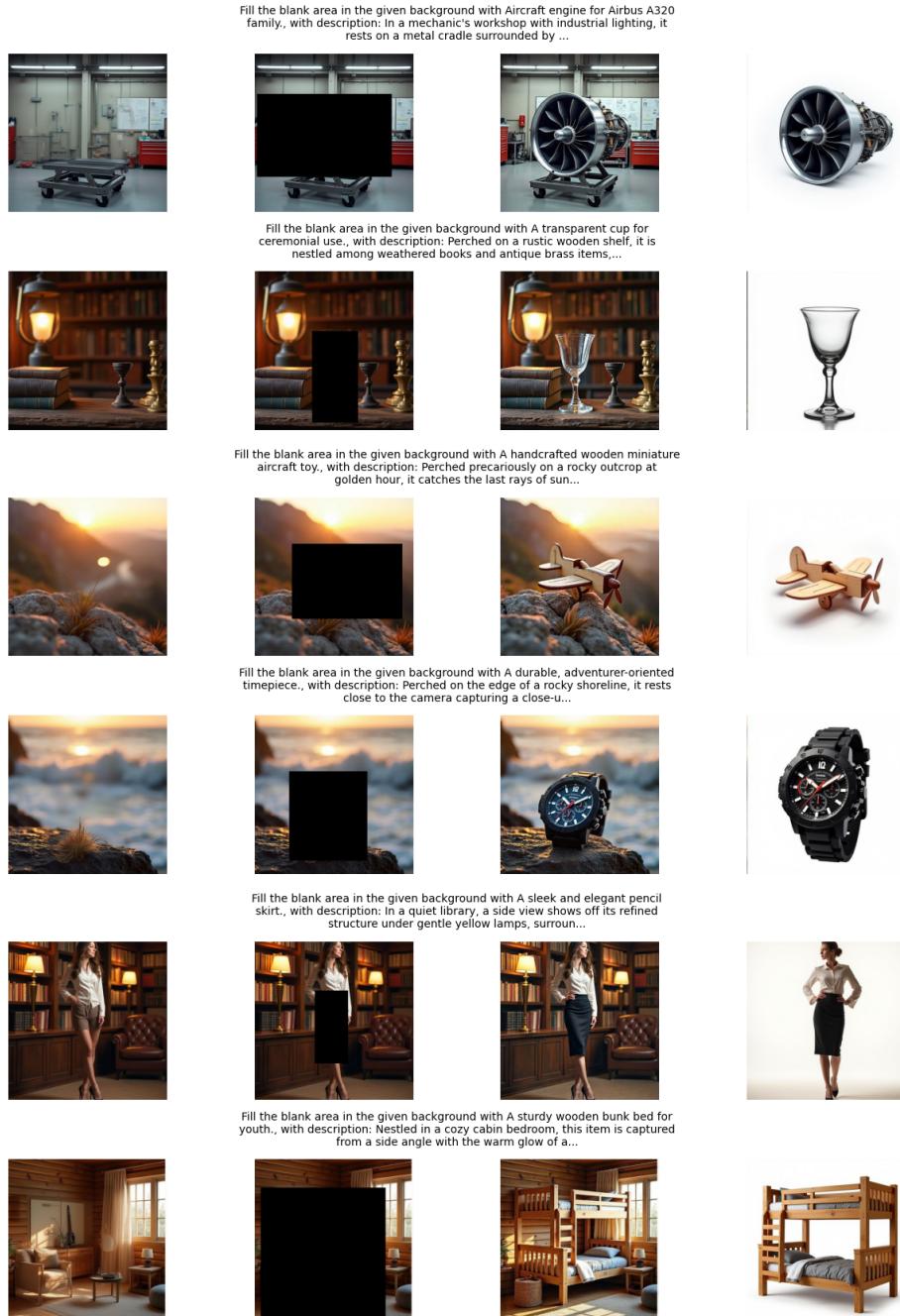


Figure 14: More results of OIN

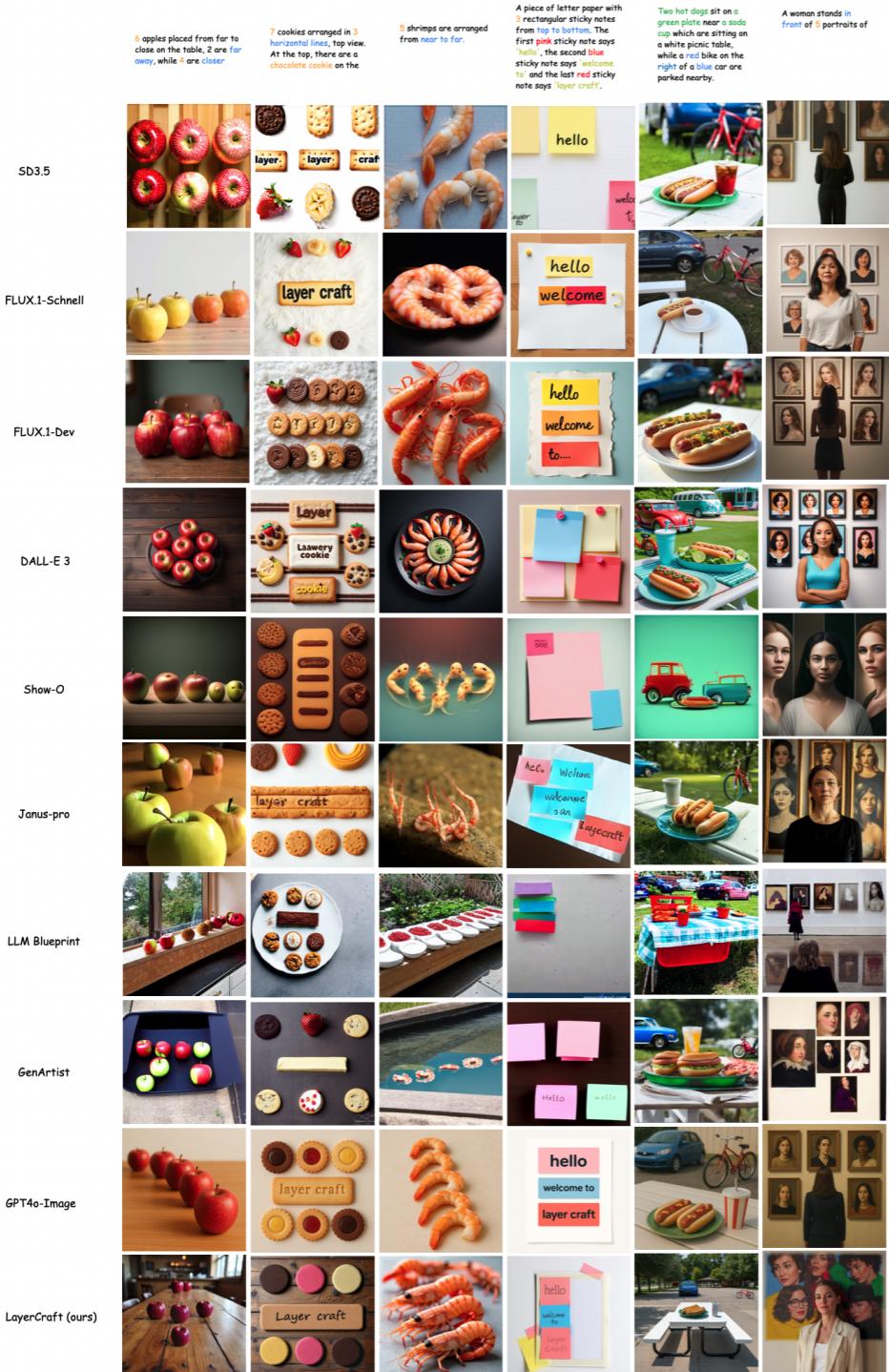


Figure 15: More comparisons with state-of-the-art models. We also include the examples used in the main paper and make them larger for details.

USER: I want to decorate this room.

LAYERCRAFT COT THINKING PROCESS: This image shows a minimalist, modern living room with a clean, warm aesthetic. Let me identify the objects and describe the background in JSON format.

```
{  
  "objects": [  
    {  
      "name": "reading nook",  
      "description": "A built-in wooden seating area with  
                     dark cushions and white pillows nestled within  
                     the wooden shelving unit"  
    },  
    {  
      "name": "pendant light",  
      "description": "A circular, disc-shaped hanging  
                     light fixture with a diffuse glow suspended  
                     from the ceiling"  
    },  
    {  
      "name": "area rug",  
      "description": "A large, neutral-toned rectangular  
                     rug covering part of the wooden floor"  
    },  
    {  
      "name": "shelving unit",  
      "description": "Built-in light wood shelving  
                     spanning the back wall with various decorative  
                     items displayed"  
    },  
    {  
      "name": "decorative objects",  
      "description": "Small sculptures, books, and framed  
                     artwork arranged sparsely on the shelves"  
    },  
    {  
      "name": "sliding glass doors",  
      "description": "Large floor-to-ceiling glass doors/  
                     windows on the left side offering views to the  
                     outdoors"  
    }  
  ],  
  "background": {  
    "description": "A spacious, minimalist room with  
                  light wooden flooring, white ceiling, and light  
                  wood wall paneling. The room features built-in  
                  shelving along the back wall and large sliding  
                  glass doors that allow natural light to flood the  
                  space. The viewpoint is from the center of the  
                  room looking toward the back wall with the built-  
                  in shelving and seating area, with the glass  
                  doors visible on the left side.",  
    "included_elements": ["wooden flooring", "white  
                          ceiling", "light wood wall paneling", "natural  
                          lighting"]  
  }  
}
```

The shelving unit and built-in seating are considered separate objects but could be viewed as part of the architectural design. I've included the structural elements like flooring, ceiling, and wall paneling in the background description since they define the space rather than being foreground objects.

Looking at the image, I can suggest several items that could be added to enhance the minimalist space while maintaining its clean aesthetic. Let me think about what would complement this room:

```
{
  "suggested_additions": [
    {
      "name": "coffee table",
      "description": "A low, rectangular wooden coffee table with clean lines, placed in front of the reading nook on the area rug"
    },
    {
      "name": "side table",
      "description": "A small, round side table in light wood or brass next to the reading nook"
    },
    {
      "name": "floor lamp",
      "description": "A slim, architectural floor lamp with warm lighting placed in the corner opposite the glass doors"
    },
    {
      "name": "accent chair",
      "description": "A minimalist armchair in a complementary neutral tone (tan leather or light gray fabric) positioned at an angle to the reading nook"
    },
    {
      "name": "throw blanket",
      "description": "A textured throw blanket in a subtle pattern or muted color draped over one corner of the reading nook"
    },
    {
      "name": "indoor plant",
      "description": "A tall potted plant like a fiddle leaf fig or snake plant placed near the glass doors to bring in natural elements"
    },
    {
      "name": "decorative bowl",
      "description": "A sculptural bowl or vessel on one of the open shelves to add visual interest"
    }
  ]
}
```

These additions would enhance the functionality and warmth of the space while respecting its minimalist design language and natural materials.

Analyzing the image based on the provided grid regions, I'll determine suitable placement areas for the suggested objects while considering perspective and available space:

```
{
  "object_placements": [
    {
      "object": "coffee table",
      "suitable_regions": ["center", "between center and center-right", "between lower center and center", "between lower center-right and center"]
    }
  ]
}
```

```

    "reasoning": "The central floor area on the rug
      provides ample space for a coffee table,
      positioned in front of the reading nook"
},
{
  "object": "side table",
  "suitable_regions": ["center-right", "between
    center and center-right"],
  "reasoning": "A small side table would fit well
    beside the reading nook on the right side"
},
{
  "object": "floor lamp",
  "suitable_regions": ["lower left", "between lower
    left and center-left"],
  "reasoning": "The corner area opposite the glass
    doors provides good placement for a tall floor
    lamp without blocking views or pathways"
},
{
  "object": "accent chair",
  "suitable_regions": ["lower center-left", "between
    lower center-left and center", "center-left", "between
    center-left and center"],
  "reasoning": "An accent chair would fit well in
    this area, creating a conversation zone with
    the reading nook while maintaining open space"
},
{
  "object": "throw blanket",
  "suitable_regions": ["center-right"],
  "reasoning": "The throw blanket would be placed on
    the reading nook which is already in this
    region"
},
{
  "object": "indoor plant",
  "suitable_regions": ["upper left", "between upper
    left and center-left"],
  "reasoning": "Near the glass doors to receive
    natural light while adding greenery to that
    corner of the room"
},
{
  "object": "wall art",
  "suitable_regions": ["upper center-left", "upper
    center", "between upper center-left and center",
    "between upper center and center"],
  "reasoning": "The visible wall space opposite the
    shelving unit would accommodate wall art while
    maintaining visual balance"
},
{
  "object": "decorative bowl",
  "suitable_regions": ["center-right", "between
    center and center-right"],
  "reasoning": "Could be placed on one of the
    existing shelves in the built-in unit"
}
],
"spatial_considerations": {
  "viewing_perspective": "The image is taken from a
    position looking toward the back wall with the
    built-in shelving, with the glass doors on the
    left side",
}

```

```

        "floor_space": "The central and lower areas have the
                      most available floor space for furniture
                      placement",
        "traffic_flow": "Pathways should be maintained
                      between the entrance (presumed to be behind the
                      viewing position) and the glass doors",
        "focal_points": "The reading nook and shelving unit
                      already serve as focal points, so additional
                      elements should complement rather than compete
                      with them"
    }
}

```

This analysis considers both the physical space available in each region and the design principles of balance, flow, and functionality while maintaining the minimalist aesthetic of the room.

Based on our analysis, here are the bounding boxes and prompts for each suggested object, ordered from far to close in terms of generation order:

```

{
  "objects": [
    {
      "type": "indoor plant",
      "position": "upper left",
      "generation_order": 1,
      "prompt": "A tall, elegant fiddle leaf fig plant
                  with large, glossy green leaves in a minimalist
                  ceramic pot, placed near large windows to
                  receive natural light, rendered in a clean,
                  contemporary style that complements modern
                  interior design.",
      "bounding_box": [50, 150, 150, 350]
    },
    {
      "type": "decorative bowl",
      "position": "center-right",
      "generation_order": 2,
      "prompt": "A sculptural, handcrafted ceramic bowl
                  in a soft matte finish with subtle organic
                  texture and asymmetrical form, placed on an
                  open shelf, rendered in a minimalist
                  Scandinavian style.",
      "bounding_box": [420, 220, 470, 260]
    },
    {
      "type": "throw blanket",
      "position": "center-right",
      "generation_order": 3,
      "prompt": "A soft, textured throw blanket in muted
                  beige with subtle geometric pattern, casually
                  draped over the corner of the built-in seating
                  area, rendered in a warm, inviting style that
                  adds comfort to the minimalist space.",
      "bounding_box": [420, 250, 500, 300]
    },
    {
      "type": "side table",
      "position": "between center and center-right",
      "generation_order": 4,
      "prompt": "A small, round side table with slender
                  brass legs and a light oak top, positioned
                  beside the reading nook at perfect arm's reach
                  height, rendered in a clean, architectural
                  style."
    }
  ]
}

```

```

        "bounding_box": [350, 250, 420, 320]
    },
    {
        "type": "floor lamp",
        "position": "between lower left and center-left",
        "generation_order": 5,
        "prompt": "A tall, architectural floor lamp with a
            slender brushed brass stem and minimal white
            shade casting a warm glow, positioned in the
            corner of the room, rendered in a contemporary
            style that emphasizes clean lines.",
        "bounding_box": [70, 320, 150, 450]
    },
    {
        "type": "accent chair",
        "position": "center-left to between center-left and
            center",
        "generation_order": 6,
        "prompt": "A minimalist lounge chair with gentle
            curves, upholstered in natural tan leather with
            a light wooden frame, positioned at an
            inviting angle to create conversation space,
            rendered in a Scandinavian modern style.",
        "bounding_box": [100, 250, 250, 380]
    },
    {
        "type": "coffee table",
        "position": "center to between lower center and
            center",
        "generation_order": 7,
        "prompt": "A low, rectangular coffee table with
            clean lines and rounded corners, crafted from
            light oak with a subtle grain pattern and
            minimalist design, positioned centrally on the
            area rug, rendered in a contemporary style that
            balances form and function.",
        "bounding_box": [200, 300, 350, 400]
    }
]
}

```

This arrangement considers depth perception (generating background objects first), spatial relationships, and the existing elements in the room. The bounding boxes are designed to place objects in appropriate locations while maintaining a balanced, functional layout that enhances the minimalist aesthetic of the space.

USER: I've done the previous decorating and want to put this cute teddy bear in.

LAYERCRAFT: Looking at the new item you want to add to the room, I can help place the teddy bear in the previously discussed space.

```
{
    "additional_object": [
        {
            "type": "teddy bear",
            "position": "lower center-right",
            "generation_order": 8,
            "prompt": "A small, golden-brown teddy bear with a
                smiling face and soft plush texture, sitting
                upright on one of the built-in shelves, adding
                a touch of warmth and playfulness to the
                minimalist space.",
            "bounding_box": [290, 300, 480, 490]
        }
    ]
}
```

}

These placements would work well with your previous decorating plans.