Beyond Atomic Layouts: Compositional Design Understanding with Vision-Language Models

Yiyang Huang[1,2⋆], Zhaowen Wang[2†], Simon Jenni[2], Jing Shi[2], Yitian Zhang[1], Yizhou Wang[1], and Yun Fu[1]

[1] Northeastern University, Boston, USA
[2] Adobe Research, San Jose, USA
https://hukcc.github.io/Beyond-Atomic-Layouts/

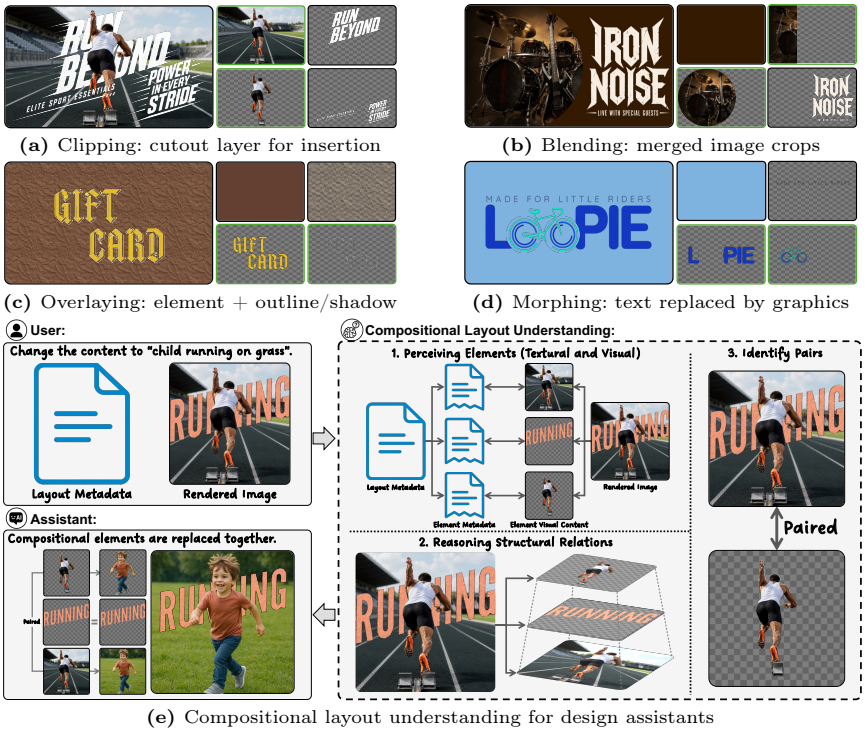
Abstract. Layout understanding, or the interpretation of element organization, is essential for document analysis, user interface (UI) creation, and graphic design. While recent vision-language models (VLMs) excel at interpreting atomic layouts composed of independent elements, they struggle with compositional layouts that require reasoning over visually entangled elements within hierarchical multi-layer structures. In this paper, we introduce a new task, compositional layout understanding, and present CoDeLayout, a VQA dataset of ∼20K real-world multi-layer layouts annotated with compositional element pairs and design intent. Through empirical analysis on CoDeLayout, we identify two key challenges for existing VLMs: semantic drift between textual metadata and visual content, and structural ambiguity in hierarchical inter-element relationships. To address these challenges, we propose MASON, a post-training paradigm that integrates multimodal alignment (MA) and structural perception (SP). MA enhances element interpretation by grounding metadata-defined elements to their visual counterparts, mitigating semantic drift, while SP models layer-aware inter-element spatial relationships to improve hierarchical understanding and reduce structural ambiguity. Experiments reveal substantial gaps in existing VLMs: even the strongest baseline, GPT-o3, achieves only 79.68% accuracy, whereas Qwen2.5-VL 7B with MASON reaches 91.66%. Notably, MASON surpasses full-data Direct Finetune using only 30% of the training data and scales better with additional data.

Keywords: Compositional Layout · Graphic Design · Post-training

1 Introduction

Layout understanding, the ability to perceive structural organization and relationships among elements such as text, images, and graphics, is a fundamental

⋆ This work was done during an internship at Adobe Research.
† Project lead.



(a) Clipping: cutout layer for insertion **(b)** Blending: merged image crops **(c)** Overlaying: element + outline/shadow **(d)** Morphing: text replaced by graphics **(e)** Compositional layout understanding for design assistants

Fig. 1: Compositional layouts in graphic design. **(a–d)** show representative compositional layouts, with green boxes marking key elements. **(a)** Clipping extracts a region from a base image as a separate layer, allowing insertion of new elements between the base and the cutout; **(b)** Blending integrates different crops of the same image into stylized shapes for aesthetic coherence; **(c)** Overlaying combines an element with its outline or shadow to emphasize visual hierarchy; and **(d)** Morphing replaces text with graphics to create expressive visual effects. **(e)** Compositional layout understanding is critical for design assistants, as handling such layouts often requires coordinated edits across interrelated elements to faithfully implement user intent.

capability for intelligent systems in document analysis [18], user interface (UI) creation [14], and graphic design [42].

Recent VLMs have shown strong performance in layout interpretation. In document understanding, models such as the LayoutLM series [18, 53, 54] integrate semantic and layout representations through 2D position embeddings, enabling fine-grained reasoning over individual document elements such as headers, tables, and forms. In user interface (UI) understanding, datasets such as ScreenQA [15] enable models like CogAgent [14] to learn navigation in graphical user interfaces (GUIs) through interaction-related visual question answering (VQA) at the level of individual interface elements. In graphic design, recent approaches address tasks such as extracting quantitative information from infographics [38] and analyzing rhetorical intent in advertisements [36] by modeling

grouping relationships among individual layout elements. However, existing research is primarily developed for atomic layouts composed of independent elements, leaving compositional structures underexplored.

In contrast to atomic layouts, which typically involve clearly separated elements with explicit boundaries and rely on 2D positional encodings to model reading order, compositional designs involve visual entanglement among multiple elements within hierarchical structures and require layer-aware spatial reasoning to capture inter-element relationships. Figure 1 (a-d) illustrates representative composition types, including *clipping* for inter-layer insertion, *blending* image crops into stylized shapes, *overlaying* an element with its outline, and *morphing* text for visual effects. Furthermore, as shown in Figure 1e, understanding compositional layouts is essential for enabling design assistants to perform editing operations correctly. This, in turn, requires the underlying VLMs to interpret elements under visual and semantic ambiguity arising from multi-element entanglement and to reason over layer-aware inter-element relationships. Accordingly, we introduce compositional layout understanding as a new task for VLMs and present CoDeLayout, a VQA dataset designed to support it. Comprising approximately 20K real multi-layer design layouts, CoDeLayout provides high-fidelity rendered images paired with element-level metadata and QA-style annotations that capture compositional element pairs and their associated design intents.

Empirical analysis on CoDeLayout reveals that existing VLMs struggle with compositional layouts due to two key challenges: *semantic drift* and *structural ambiguity*. Semantic drift emerges from the visual entanglement of elements, where VLMs fail to align textual metadata with visual content, leading to incorrect interpretation of element semantics and design intent. Structural ambiguity stems from the hierarchical organization of compositional elements, where VLMs struggle to interpret layer-aware spatial structures and consequently fail to capture inter-element relationships.

To address these challenges, we propose **MASON**, a post-training paradigm integrating **M**ultimodal **A**lignment (MA) and **S**tructural percepti**ON** (SP). MA mitigates semantic drift by aligning metadata-defined elements to their visual counterparts, enabling coherent interpretation of design attributes and visual semantics. In parallel, SP addresses structural ambiguity by extracting layer-aware spatial relationships among elements that encode their positional and hierarchical dependencies.

Experiments reveal substantial gaps in existing VLMs on compositional layout understanding: even the strongest baseline, GPT-o3 with image-based reasoning capability, achieves only 79.68% accuracy. In contrast, Qwen2.5-VL 7B with MASON achieves 91.66% accuracy and consistently improves performance across diverse compositional design categories, including overlaying, clipping, blending, and morphing. Furthermore, MASON demonstrates strong data efficiency, surpassing full-data Direct Finetune while using only 30% of the training data, and shows better scalability as training data increases.

Our main contributions are summarized as follows:

- We introduce compositional layout understanding as a new task and construct CoDeLayout, the first VQA dataset for this setting, comprising ∼20K real multi-layer layouts with explicit annotations of compositional element pairs and their design intent.
- Through empirical analysis on CoDeLayout, we identify two core challenges, namely *semantic drift* and *structural ambiguity*, explaining why existing VLMs struggle with compositional layouts.
- We propose MASON, a post-training paradigm that mitigates semantic drift via multimodal alignment and addresses structural ambiguity through layer-aware structural perception.
- Experiments reveal substantial performance gaps in existing VLMs and validate the effectiveness of MASON. Notably, MASON surpasses full-data Direct Finetune using only 30% of the training data and scales more effectively with increasing data.

2 Related Work

We review related work in three areas: vision-language models as the foundation for layout reasoning, layout generation methods for design synthesis, and layout understanding approaches primarily focused on atomic layouts.

2.1 Vision Language Models

Recent advances in vision-language models (VLMs) [17,34,35,55–57] have substantially improved multimodal understanding by integrating large language models (LLMs) [2,7–9,11,49] with visual encoders such as CLIP [45]. Flamingo [1] introduced interleaved vision-language modeling for open-ended reasoning, while BLIP-2 [29] bridged frozen vision and language backbones through a Q-Former. Instruction-tuned models such as LLaVA [31,32] and Qwen-VL [3,50] further improved visual grounding and cross-modal reasoning. Building on these advances, spatially aware VLMs such as Qwen2.5-VL [5], Qwen3-VL [4], InternVL-3.5 [51], and LLaVA-OneVision [25] further advance dense localization and object-level grounding. Collectively, these advances provide a strong foundation for layout perception and reasoning.

2.2 Layout Generation

Layout generation aims to synthesize new designs from learned structural patterns and has progressed across documents, user interfaces, and graphic design. For documents, methods such as LayoutVAE [23], LayoutGAN++ [24], LayoutDM [20], and Text2Poster [22] generate layouts that follow logical and visual hierarchies. For user interfaces, Pix2Code [6] and Design2Code [46] synthesize coherent layouts from textual descriptions or sketches. In graphic design, datasets including Magazine [52], GenPoster-100K [13], DesignerIntention [21], Crello [47], and MLTD [44] support multi-layer layout modeling, enabling models such as LayoutGAN [27], LayoutTransformer [12], COLE [21], and ART [44] to learn geometric constraints and generate layouts under multimodal guidance.

Metadata $\mathcal{M}$:
- Id: *Design-67*
- Canvas-size: *[1080,1080]*
- Elements:
  - Element-id: *element-1*
  - Element-type: *image*
  - Parent-element-id : *N/A*
  - Opacity: *1.0*
  - Bounds: *width, height*
  - Placement: *xscale, yscale, rotation, tx, ty*
  - Frame: *topLeft, topRight, bottomRight, bottomLeft*
  - ZIndex: *1*

QA Pair:

Question $\boldsymbol{Q(e^q)}$: The element with ID $\mathcal{M}_{e^q}[id]$ is part of a compositional pair. What is the ID of the other element it is composed with, and what is their compositional purpose?

Answer $\boldsymbol{A(e^{comp})}$: The other element's ID is $\mathcal{M}_{e^{comp}}[id]$. The compositional purpose is [the specific compositional purpose of this pairing].

Fig. 2: Data format of a compositional design. Each instance includes a rendered layout image, element-level metadata, and QA-style annotations specifying compositional element pairs and their associated design intents.

2.3 Layout Understanding

Layout understanding aims to perceive and reason over the spatial and semantic organization of elements and is central to document analysis, UI creation, and graphic design. In documents, datasets such as PubLayNet [59] and DocLayNet [43] support models like LayoutLM [18, 53, 54] and DiT [28], which encode semantic and layout features with 2D positional embeddings. For UIs, RICO [10] and ScreenQA [15] enable navigation in graphical user interfaces (GUIs) and interaction-related visual question answering (VQA) with models such as CogAgent [14]. In graphic design, prior work addresses tasks including infographic question answering [37, 38] and rhetorical analysis in advertisements [36] by modeling grouping relationships among individual elements.

Although effective for atomic layouts, these datasets and methods assume independent elements and rely on 2D positional encodings, limiting their ability to model hierarchical and layer-aware interactions among visually entangled elements. Such interactions are fundamental to compositional layout understanding and motivate the need for a dedicated dataset and a tailored modeling paradigm.

3 Compositional Layout Understanding: Task & Dataset

3.1 Task Formulation

Compositional layout understanding aims to identify interacting elements within hierarchical layouts and interpret their design intent. As illustrated in Fig. 2, each layout instance is represented as $\mathcal{L} = (\mathcal{I}, \mathcal{M})$, where $\mathcal{I}$ is the rendered design image and $\mathcal{M} = \{\mathcal{M}_{e_1}, \mathcal{M}_{e_2}, \ldots, \mathcal{M}_{e_n}\}$ denotes the complete metadata of the design. Each $\mathcal{M}_{e_i}$ contains attributes of element e_i, including its type, spatial position (placement), and stacking order (Zindex). Given a question $Q(e^q)$ referring to a query element e^q, the model is required to identify the corresponding compositional element e^{comp} and generate a textual answer $A(e^{comp})$ explaining their design intent:

$$A(e^{comp}) = \text{VLM}(\mathcal{L}, Q(e^q)) \tag{1}$$

Table 1: Comparison of representative layout datasets. In contrast to existing datasets that primarily focus on atomic layouts or generative objectives, CoDeLayout provides multi-layer layouts with explicit annotations of inter-element compositional relationships and the design intent.

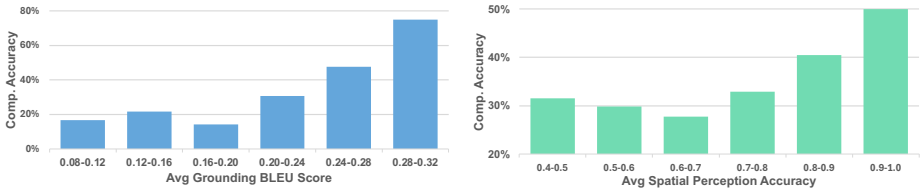
Dataset	Annotation	Structure	Domain	# Samples	# Layers
Layout Generation					
RICO [10]	N/A	Flat	UI	∼72K	N/A
Magazine [52]	N/A	Layering	Graphic	∼3.9K	N/A
GenPoster-100K [13]	N/A	Layering	Graphic	∼100K	2–20
DesignerIntention [21]	N/A	Layering	Graphic	∼100K	3–10
Crello [47]	N/A	Layering	Graphic	∼20K	2–50
MLTD [44]	N/A	Layering	Graphic	∼1M	2–50
Layout Understanding					
PubLayNet [59]	Atomic	Flat	Doc	∼360K	N/A
DocLayNet [43]	Atomic	Flat	Doc	∼80K	N/A
ScreenQA [15]	Atomic	Flat	UI	∼86K	N/A
CoDeLayout (Ours)	Compositional	Layering	Graphic	∼20K	3–50

3.2 Compositional Design Layout Dataset

To support compositional layout understanding in graphic design, we construct the Compositional Design Layout Dataset (CoDeLayout), a high-quality, multi-layer design dataset centered on element-level compositional relationships. Detailed data collection procedures are provided in the supplementary materials.

Data Format & Statistics CoDeLayout contains approximately 20K design instances spanning diverse resolutions and coherent multi-layer structures across four representative compositional types: Overlaying, Clipping, Blending, and Morphing. As illustrated in Fig. 2, each instance includes: (1) a high-fidelity rendered design image; (2) a structured JSON file with element-level metadata (`id`, `type`, `opacity`, `zIndex`, `position`, etc.); and (3) QA-style annotations $(Q(e^q), A(e^{comp}))$ specifying compositional element pairs (e^q, e^{comp}) and their associated design intents.

CoDeLayout follows the natural distribution of real-world graphic designs without resampling for category balancing. It contains 20,009 training samples and 387 test samples. In the training set, Blending dominates (56.78%), followed by Overlaying (22.83%), Clipping (17.76%), and Morphing (2.63%). To ensure reliable evaluation across all composition types, the test split slightly increases the proportion of rare categories, resulting in the following distribution: Blending (40.06%), Overlaying (29.95%), Clipping (17.75%), and Morphing (12.24%). All test samples are manually verified to ensure annotation correctness and prevent data leakage. In terms of resolution, CoDeLayout spans from small banners (300×60) to ultra-high-resolution prints (7200×14400), covering common social media formats (e.g., 1080×1080, 1080×1920) and standard print sizes (e.g., A4 at 2480×3507). The designs contain an average of 17.4 layers, with 99.8% containing fewer than 45 layers.



(a) Semantic Drift. Grounding performance (x-axis: average BLEU score per layout) vs. compositional element identification accuracy (y-axis).

(b) Structural Ambiguity. Spatial relation perception accuracy (x-axis) vs. compositional element identification accuracy (y-axis).

Fig. 3: Impact of semantic drift (a) and structural ambiguity (b) on compositional layout understanding. Higher grounding performance and spatial relation perception accuracy consistently correspond to better compositional element identification accuracy, indicating that multimodal alignment and structural perception are essential for compositional layout understanding.

Comparison with Existing Layout Datasets Table 1 compares representative layout datasets across understanding and generation domains. Document and UI datasets, such as PubLayNet [59], DocLayNet [43], ScreenQA [15], and RICO [10], primarily target atomic layouts with flat or tree-structured representations, supporting structural parsing and function-level reasoning. Graphic design datasets, including Crello [47], GenPoster-100K [13], and MLTD [44], provide multi-layer layouts for generation but do not explicitly model inter-element compositional relationships.

In contrast, CoDeLayout is the first dataset dedicated to compositional layout understanding. It integrates high-fidelity rendered designs, element-level metadata, and explicitly formulated QA-style annotations that capture complex compositional element pairs and their associated design intent.

3.3 Challenges in Compositional Layout Understanding

Compositional layouts introduce structural complexity beyond atomic layouts, requiring accurate element interpretation under visual entanglement (*semantic drift*) and layer-aware modeling of hierarchical relationships among elements (*structural ambiguity*). To examine these challenges, we analyze a strong VLM, GPT-4o, on CoDeLayout.

Semantic Drift Semantic drift arises from the visual entanglement of elements, as VLMs fail to align textual metadata with visual content, resulting in misinterpretation of element semantics and degraded compositional understanding.

To empirically examine this issue, we evaluate GPT-4o's ability to ground metadata-defined elements in the rendered layout image. For each layout, element crops are fed to GPT-4o with a fixed descriptive prompt ("What is element A?") to generate reference descriptions. During evaluation, the model receives the rendered layout image together with its metadata and is asked to describe

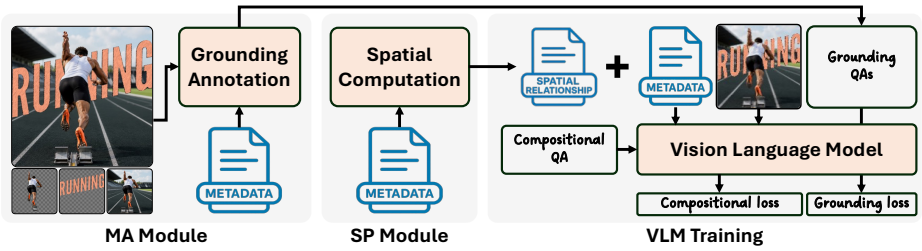


Fig. 4: Overview of MASON. MASON mitigates semantic drift and structural ambiguity through multimodal alignment (MA) and structural perception (SP). MA introduces an element grounding objective to align textual metadata with visual content under visual entanglement. SP augments metadata with layer-aware spatial relationships, enabling accurate modeling of hierarchical inter-element structures.

each element. Performance is measured by the average BLEU score between generated and reference descriptions across elements within each layout.

As shown in Figure 3a, samples with higher grounding scores consistently achieve better compositional identification accuracy. This result indicates that grounding capability directly impacts compositional layout understanding, highlighting multimodal alignment as a critical factor.

Structural Ambiguity Structural ambiguity arises from the hierarchical organization of compositional elements, where VLMs struggle to interpret layer-aware spatial structures, limiting their ability to model inter-element relationships.

To empirically examine this issue, we evaluate GPT-4o's ability to perceive layer-aware spatial relationships through a spatial relation perception task. For each layout, 30 element pairs are randomly sampled to construct yes/no questions about structural relationships (e.g., "Is element A above element B?"). During evaluation, the model receives the rendered layout image together with its metadata and answers these questions. Performance is measured by the average accuracy across the sampled element pairs within each layout.

As shown in Figure 3b, higher spatial relation perception accuracy consistently corresponds to better compositional identification accuracy. This indicates that structural perception directly affects compositional layout understanding.

4 MASON: A Baseline Post-training Paradigm

Our analysis in Section 3.3 shows that semantic drift and structural ambiguity hinder accurate element interpretation and inter-element relationship modeling, limiting VLMs' compositional layout understanding. To address these challenges, we propose MASON as a baseline paradigm for compositional layout understanding that integrates multimodal alignment (MA) and structural perception (SP) into VLM post-training, as illustrated in Figure 4.

Table 2: Prompt template $\mathcal{P}_{\text{ground}}$ for generating element grounding QA pairs in multimodal alignment.

Element Grounding Prompt $\mathcal{P}_{\text{ground}}$

Inputs:
- Full design image: $\mathcal{I}$
- Layout metadata: $\mathcal{M} = \{\mathcal{M}_{e_1}, \mathcal{M}_{e_2}, \ldots, \mathcal{M}_{e_n}\}$
- Element ID: `{element_id}`
- Image crop and metadata of the element: $(\mathcal{I}_e, \mathcal{M}_e)$

Grounding Question $Q^g(e_i)$:
"What is the element with ID `{element_id}` and where is it located in the complete design?"

Example Answer:
"The element with ID `{element_id}` is a [description]. It is located in the [position] of the design. [Optional: Additional identifying details]."

Please generate the answer based on both the element's metadata and visual content, as well as the global layout context.

4.1 Multimodal Alignment (MA)

To address semantic drift, MA introduces an element grounding objective during VLM post-training, encouraging alignment between textual metadata and corresponding visual content under visual entanglement.

The grounding objective is constructed from 1K layouts sampled from the training split of CoDeLayout. Each instance follows the layout formulation $\mathcal{L} = (\mathcal{I}, \mathcal{M})$ and is augmented with grounding QA supervision (Q^g, A^g).

Specifically, for each layout, 20% of elements are randomly sampled. For each selected element e_i, an off-the-shelf VLM $\mathcal{G}$ generates a grounding QA pair. Given a layout $\mathcal{L}$, together with the element crop $\mathcal{I}_{e_i}$, its metadata $\mathcal{M}_{e_i}$, and a grounding question $Q^g(e_i)$ ("What is the element with ID `{element_id}` and where is it located in the complete design?"), $\mathcal{G}$ produces an answer $A^g(e_i)$ describing its semantic identity and spatial placement:

$$A^g(e_i) = \mathcal{G}\big(\mathcal{P}_{\text{ground}}(\mathcal{L}, \mathcal{I}_{e_i}, \mathcal{M}_{e_i}, Q^g(e_i))\big), \tag{2}$$

where $\mathcal{P}_{\text{ground}}$ denotes the grounding prompt template (Table 2).

4.2 Structural Perception (SP)

To address structural ambiguity, SP incorporates layer-aware spatial relationships into VLM post-training, enabling accurate modeling of hierarchical inter-element relationships.

Given a layout $\mathcal{L}$ with metadata $\mathcal{M} = \{\mathcal{M}_{e_1}, \ldots, \mathcal{M}_{e_n}\}$, where each element e_i is annotated with attributes such as bounding box, stacking order, and type, SP derives spatial relationships between the query element e^q, referenced in the compositional layout understanding question $Q(e^q)$, and every other element e_i. These relationships are encoded in an additional metadata

Table 3: Schema of layer-aware inter-element spatial relationships used for structural perception.

Spatial Relationship Schema

`spatialRelationship: {`
`  hasOverlap: <True | False>,  overlapPercentage: <float>,`
`  containment: <contain | contained | none>,  centerDistance: <float>,`
`  relativePosition: <left | right | above | below>,  layerOrder: <above_given | below_given>`
`}`

field `spatialRelationship`, which includes six properties: overlap status and ratio, containment type, center distance, directional relation, and relative stacking order (Table 3). The enriched metadata for each element is defined as:

$$\mathcal{M}'_{e_i} = \mathcal{M}_{e_i} \cup \texttt{spatialRelationship}(e_i, e^q). \tag{3}$$

Embedding these spatial attributes into the metadata provides explicit geometric context, facilitating interpretation of layout hierarchy and inter-element structure.

4.3 Integration into VLM Post-training

During post-training, multimodal alignment and structural perception are incorporated through metadata augmentation and joint supervision from compositional and grounding QA tasks.

Specifically, the original metadata $\mathcal{M}$ is augmented with layer-aware spatial relationships, resulting in $\mathcal{M}' = \{\mathcal{M}'_{e_1}, \ldots, \mathcal{M}'_{e_n}\}$, where each $\mathcal{M}'_{e_i}$ includes the `spatialRelationship` relative to the query element e^q. The VLM is optimized on a mixture of compositional and grounding QA samples:

$$\hat{A} = \mathrm{VLM}\big((\mathcal{I}, \mathcal{M}'),\ Q\big), \tag{4}$$

where Q denotes either a compositional question $Q(e^q)$ or a grounding question $Q^g(e_i)$ depending on the training instance. The output is optimized using cross-entropy loss against the corresponding ground-truth answer.

At inference time, layer-aware metadata augmentation is retained, while evaluation focuses solely on the compositional layout understanding task.

5 Experiments

5.1 Baselines and Metrics

Baselines. We evaluate heuristic methods, open-source models, and proprietary large-scale models on CoDeLayout to assess compositional layout understanding in current VLMs under zero-shot and default inference settings. Heuristic baselines include Max-Overlap, which selects the element with the highest IoU with

the query e^q, and Nearest-Neighbor, which selects the element with the nearest centroid. Open-source models (7B scale) include Qwen2.5-VL [5], Qwen3-VL [4], InternVL-3.5 [51], and LLaVA-OneVision [26]. Proprietary models include GPT-4o [19], GPT-5 [39], GPT-o3 [40], Gemini-2.5-flash [48], and Gemini-2.5-pro [48]. Models with explicit reasoning capabilities (GPT-5, GPT-o3, and Gemini-2.5-pro) are evaluated under their default medium reasoning configuration.

For post-training comparison, we further evaluate two Qwen2.5-VL [5] 7B variants: Direct Finetune, finetuned solely on compositional QA data, and MASON, optimized with multimodal alignment and structural perception.

Metrics. We report Accuracy, GPT-Score (0–10), BLEU, and ROUGE. Accuracy evaluates correct identification of the compositional element e^{comp} given the query e^q, reflecting compositional element selection. For generated explanations, BLEU [41] and ROUGE [30] measure lexical overlap with reference answers, while GPT-Score [58] assesses semantic alignment with the annotated design intent. Detailed scoring prompts are provided in the supplementary materials.

5.2 Implementation Details

All models are evaluated under consistent prompting, decoding, and input configurations. Open-source VLMs (7B scale) are tested using their official instruction-following interfaces in zero-shot mode, while proprietary models are accessed via API under default inference settings. Input images are resized to 336×336 and paired with structured JSON metadata. All models use greedy decoding with a maximum output length of 256 tokens to ensure deterministic and parsable responses.

Our post-training variants, Direct Finetune and MASON, are implemented on Qwen2.5-VL (7B) [5] and trained on CoDeLayout. Training uses the AdamW optimizer [33] with a learning rate of 3×10^{-5}. LoRA adapters [16] ($r = 4$, $\alpha = 8$) are applied to the attention projection layers (`q`/`v`). The context length is set to 4096 tokens, and post-training is conducted for 3 epochs with a global batch size of 64 on 8 A100 GPUs. For multimodal alignment, GPT-4o is used to generate grounding QA supervision. In experiments with type-balanced subsampling, abundant types (e.g., Overlaying, Blending, and Clipping) are sampled less frequently than scarce types (e.g., Morphing).

5.3 Comparison on CoDeLayout

As shown in Table 4, heuristic baselines perform poorly across categories, indicating that simple geometric priors (e.g., overlap or centroid proximity) are insufficient for compositional element identification. Open- and closed-source VLMs also exhibit clear performance gaps under zero-shot settings. Even the strongest baseline, GPT-o3, achieves 79.68% weighted accuracy and 76.28% average accuracy, underscoring the difficulty of modeling visually entangled and layer-aware structures.

Table 4: Performance on CoDeLayout across four compositional categories. Each cell reports Accuracy (GPT-Score/BLEU/ROUGE). Accuracy (↑) evaluates compositional element identification, while GPT-Score/BLEU/ROUGE (↑) measure explanation quality. **Bold** and underline indicate the best and second-best results. Weighted Accuracy is computed using test sample proportions, and Average Accuracy is the unweighted mean across categories. The results highlight challenges in compositional layout understanding for existing VLMs and MASON's effectiveness and data efficiency.

Model	Overlaying	Clipping	Blending	Morphing	Weighted Acc.	Average Acc.
Heuristic Baselines						
Max-Overlap	61.74 (N/A)	69.70 (N/A)	23.08 (N/A)	36.17 (N/A)	44.27	47.67
Nearest-Neighbor	58.26 (N/A)	62.12 (N/A)	23.72 (N/A)	38.30 (N/A)	42.44	45.60
Open-Source VLMs						
Qwen2.5-VL	66.09 (3.51/3.65/23.94)	37.88 (2.85/1.16/18.94)	53.85 (3.37/2.48/22.28)	36.17 (3.22/2.16/21.42)	52.60	48.49
Qwen3-VL	86.09 (4.52/7.63/32.77)	71.21 (3.85/3.81/26.12)	79.49 (4.11/4.59/28.55)	55.32 (3.49/2.22/23.04)	77.08	73.02
InternVL-3.5	84.35 (4.02/5.66/28.84)	54.55 (2.93/1.29/19.49)	70.51 (3.52/2.74/23.38)	42.55 (3.11/1.70/20.48)	68.48	62.99
LLaVA-OneVision	56.52 (3.41/2.90/23.21)	21.21 (2.72/0.71/17.59)	60.90 (3.24/1.74/21.36)	29.79 (2.89/0.97/18.82)	48.95	42.10
Closed-Source VLMs						
GPT-4o	83.48 (4.91/10.67/36.34)	36.36 (4.02/3.22/28.05)	63.46 (4.66/6.22/34.00)	65.96 (4.22/5.88/29.89)	65.10	62.31
GPT-5	89.57 (3.62/4.55/23.95)	62.12 (2.99/1.40/19.18)	79.49 (3.25/2.20/21.25)	56.92 (3.51/1.98/23.97)	76.56	71.62
GPT-o3	93.91 (3.35/3.26/22.08)	63.64 (2.81/1.25/18.46)	79.49 (3.12/2.25/19.93)	68.09 (3.46/1.87/23.73)	79.68	76.28
Gemini-2.5-flash	81.74 (4.36/6.88/31.18)	53.03 (3.54/2.96/23.87)	72.44 (3.78/3.61/25.14)	55.32 (3.81/2.85/25.66)	69.79	65.63
Gemini-2.5-pro	90.43 (4.02/4.92/28.48)	45.45 (3.77/2.99/25.79)	75.00 (3.89/3.53/26.33)	59.57 (3.79/3.15/25.70)	72.65	67.61
Ours						
Direct Finetune (Full data)	93.04 (6.42/30.87/51.63)	83.33 (4.73/10.89/35.08)	94.87 (5.14/11.69/39.53)	65.96 (4.96/11.23/37.00)	88.80	84.30
MASON (30% data)	92.17 (6.29/29.92/50.22)	86.36 (4.81/10.69/35.68)	93.59 (4.86/9.80/36.34)	**72.34** (4.69/9.95/34.44)	89.32	86.12
MASON (Full data)	**95.65** (6.62/31.87/52.99)	**90.91** (4.93/11.74/36.27)	**95.51** (5.22/11.84/39.66)	70.21 (5.13/11.60/38.77)	**91.66**	**88.07**

MASON consistently outperforms all baselines. Compared with GPT-o3, MASON (Full data) improves weighted accuracy from 79.68% to 91.66% and average accuracy from 76.28% to 88.07%, with especially large gains on structurally complex categories such as Clipping (90.91% vs. 63.64%), where layer-aware spatial modeling under visual entanglement is essential. Notably, even MASON achieves only 70.21% accuracy on the challenging Morphing category, which is characterized by strong visual-semantic disruption and limited training data (∼3%), suggesting that CoDeLayout remains challenging.

MASON also shows strong data efficiency by achieving 89.32% weighted and 86.12% average accuracy, surpassing Direct Finetune trained on the full dataset (88.80% and 84.30%) while using fewer total QA pairs (∼8K vs. 20K) under the same post-training protocol. Specifically, MASON uses only ∼6K compositional pairs (30% of the training set, obtained via type-balanced subsampling) and ∼2K grounding pairs derived from 1K layouts in the CoDeLayout training set, without relying on any external data. Additionally, the 30% subset alleviates type imbalance, yielding higher Morphing accuracy than MASON (Full data).

Table 5: Module ablation on CoDeLayout under a type-balanced setting. DF denotes Direct Finetune, MA denotes Multimodal Alignment, and SP denotes Structural Perception. Results report accuracy (↑) for each compositional category, along with Weighted and Average accuracy. Weighted accuracy accounts for category proportions, while Average accuracy is the unweighted mean. **Bold** indicates the best result in each column. Results show that MA and SP provide complementary improvements, with their combination achieving the strongest overall performance.

Model	Overlaying	Clipping	Blending	Morphing	Weighted Acc.	Average Acc.
DF	86.09	74.24	80.77	72.34	80.21	78.36
DF + MA	87.83	81.82	81.41	70.21	82.03	80.32
DF + SP	**89.57**	**83.33**	82.69	74.47	83.85	82.51
MASON	86.96	81.82	**83.97**	**80.85**	**84.11**	**83.40**

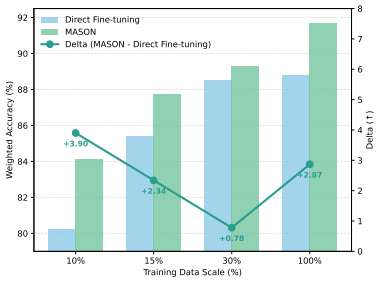


Fig. 5: Data Scale Ablation. Performance of Direct Finetune and MASON trained on 10%, 15%, 30%, and 100% of the CoDeLayout training set. The x-axis denotes training data scale, the left y-axis shows test set weighted accuracy, and the right y-axis shows accuracy gain of MASON over Direct Finetune. MASON consistently outperforms across all scales, with a clear advantage under low-resource settings.

5.4 Ablation Study

Module Ablation To assess the contribution of each module, type-balanced subsampling is applied to ensure equal sample counts across types. As shown in Table 5, both multimodal alignment (MA) and structural perception (SP) consistently improve weighted and average accuracy. Direct Finetune (DF) achieves 80.21% weighted and 78.36% average accuracy. Adding MA increases performance to 82.03% and 80.32%, indicating improved element-level alignment. Incorporating SP yields 83.85% and 82.51%, reflecting the benefit of modeling inter-element spatial structure. Combining both modules results in the full MASON model, achieving 84.11% weighted and 83.40% average accuracy, demonstrating complementary improvements in addressing semantic drift and structural ambiguity. Furthermore, the effects of MA and SP vary across composition types. For heavily occluded compositions (Overlaying/Clipping), noisy grounding may conflict with SP's structural cues, limiting the benefits of MA. In contrast, MA complements SP for Blending. For Morphing, MA alone may increase ambiguity by grounding distorted text as graphics without sufficient structural context, whereas SP helps recover the text-replacement relation.

Data Scale Ablation To further demonstrate MASON's data efficiency, we study the effect of training data scale. Direct Finetune and MASON are separately trained on 10%, 15%, 30%, and 100% of the CoDeLayout training set,

Table 6: Grounding Model Ablation. Comparable performance across grounding models suggests that the gains arise from the alignment paradigm rather than model strength.

Grounding VLM	Weighted Acc.	Average Acc.
GPT-4o	91.6%	88.1%
Qwen3-VL (direct)	90.7%	87.4%
Qwen3-VL (script)	91.1%	88.1%

Table 7: Visual Dependency Ablation. Removing visual features leads to a substantial performance drop, highlighting the importance of visual perception for compositional layout understanding.

	Weighted Acc.		Average Acc.	
Model	Visual	No Visual	Visual	No Visual
DF	88.8%	60.9%	84.3%	58.1%
MASON	91.6%	65.1%	88.0%	62.8%

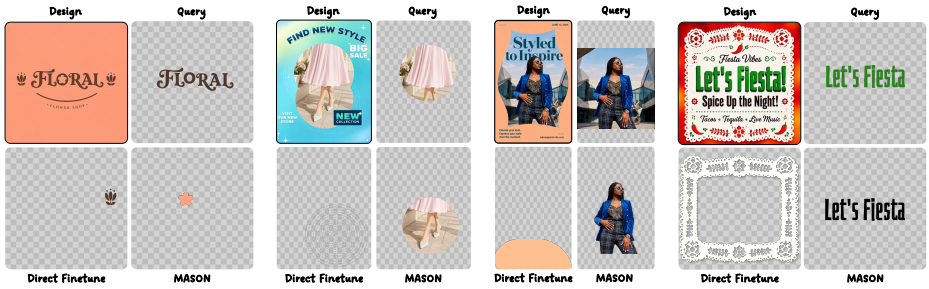
with the smaller subsets constructed via type-balanced subsampling. As shown in Fig. 5, performance improves for both methods with increasing training data, while MASON consistently outperforms Direct Finetune across all scales. Under the 10% setting, Direct Finetune suffers a notable drop in accuracy, whereas MASON maintains a clear advantage, indicating stronger data efficiency. Moreover, while Direct Finetune shows diminishing returns beyond 30% of the training data, MASON continues to benefit from additional supervision, demonstrating better scalability.

Grounding Model Ablation To determine whether the gains from MA stem from the alignment paradigm rather than the strength of a particular grounding model, we conduct an ablation using different grounding VLMs. In addition to GPT-4o, we evaluate Qwen3-VL (8B) under two settings: (1) Direct grounding, where the model generates both element descriptions and positional information; and (2) Script-based grounding, where positional information is extracted from metadata and the VLM generates only the semantic description. As shown in Table 6, script-based grounding achieves performance comparable to GPT-4o. This suggests that the improvements from MA are primarily attributable to explicit metadata–vision alignment rather than dependence on a stronger grounding model.

Visual Dependency Ablation To examine whether the model tends to overrely on metadata, we evaluate Direct Finetune and MASON with and without visual input. As shown in Table 7, removing visual features leads to a substantial performance drop for both methods. This result suggests that compositional layout understanding relies critically on visual perception and cannot be resolved through metadata or layer-aware spatial relationships alone.

6 Case Study

To illustrate MASON's advantages over Direct Finetune (DF) in compositional layout understanding, we compare their predictions on representative examples. As shown in Fig. 6, DF often struggles to localize and associate the correct



(a) Morphing **(b)** Blending **(c)** Clipping **(d)** Overlaying

Fig. 6: Case studies across four compositional categories. Each example shows the full design, the query element, and the predictions of Direct Finetune and MASON.

compositional elements when spatial or semantic cues are subtle, whereas MASON successfully identifies the underlying compositional pairs. For example, in the structurally complex Clipping case, DF fails to recognize the implicit relationship between the cutout and the base image, instead selecting a decorative element whose width matches the base image. MASON, by contrast, correctly identifies the cutout element. In the challenging Morphing case, MASON recognizes that the flower image placed over the letter "O" creates the morphing effect, whereas DF selects a decorative element aligned with the text and sharing a similar color, but unrelated to the underlying compositional relation.

7 Conclusion

This paper introduces compositional design layout understanding, a new task that challenges VLMs to interpret visually entangled elements and model layer-aware inter-element relationships within hierarchical layouts, and presents CoDeLayout, the first dataset dedicated to this setting. Through empirical analysis, we identify two core challenges: *semantic drift*, where VLMs fail to align textual metadata with visual content, and *structural ambiguity*, where VLMs struggle to capture hierarchical and spatial relationships among elements. To address these challenges, we propose MASON, a paradigm that integrates multimodal alignment and structural perception into VLM post-training. By introducing grounding-based alignment supervision and incorporating layer-aware spatial relationships, MASON enhances both element-level interpretation and inter-element structural modeling, thereby improving compositional layout understanding. Extensive experiments show that MASON consistently outperforms both open- and closed-source VLMs, including reasoning-capable models, while demonstrating strong data efficiency under limited supervision.

Future work includes developing layer-aware visual encoders tailored to multilayer compositional layouts and exploring stronger cross-modal alignment objectives to further enhance element-level grounding and relational reasoning.

References

1. Alayrac, J., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J.L., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., Simonyan, K.: Flamingo: a visual language model for few-shot learning. In: NeurIPS (2022)
2. Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J., Men, R., Ren, X., Ren, X., Tan, C., Tan, S., Tu, J., Wang, P., Wang, S., Wang, W., Wu, S., Xu, B., Xu, J., Yang, A., Yang, H., Yang, J., Yang, S., Yao, Y., Yu, B., Yuan, H., Yuan, Z., Zhang, J., Zhang, X., Zhang, Y., Zhang, Z., Zhou, C., Zhou, J., Zhou, X., Zhu, T.: Qwen technical report. CoRR **abs/2309.16609** (2023)
3. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A frontier large vision-language model with versatile abilities. CoRR **abs/2308.12966** (2023)
4. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. CoRR **abs/2511.21631** (2025)
5. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. CoRR **abs/2502.13923** (2025)
6. Beltramelli, T.: pix2code: Generating code from a graphical user interface screenshot. In: EICS. pp. 3:1–3:6. ACM (2018)
7. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: NeurIPS (2020)
8. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., et al.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna. lmsys. org (accessed 14 April 2023) **2**(3), 6 (2023)
9. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A.M., Pillai, T.S., Pellat, M., Lewkowycz, A., Moreira, E.,

Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., Fiedel, N.: Palm: Scaling language modeling with pathways. J. Mach. Learn. Res. **24**, 240:1–240:113 (2023)
10. Deka, B., Huang, Z., Franzen, C., Hibschman, J., Afergan, D., Li, Y., Nichols, J., Kumar, R.: Rico: A mobile app dataset for building data-driven design applications. In: UIST. pp. 845–854. ACM (2017)
11. Gilardi, F., Alizadeh, M., Kubli, M.: Chatgpt outperforms crowd-workers for text-annotation tasks. CoRR **abs/2303.15056** (2023)
12. Gupta, K., Lazarow, J., Achille, A., Davis, L., Mahadevan, V., Shrivastava, A.: Layouttransformer: Layout generation and completion with self-attention. In: ICCV. pp. 984–994. IEEE (2021)
13. Haoran, W., Bo, Z., Jinghui, W., Hanzhang, W., Huan, Y., Wei, J., Hao, L., Xinyan, X.: Sega: A stepwise evolution paradigm for content-aware layout generation with design prior. ICCV (2025)
14. Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Dong, Y., Ding, M., Tang, J.: Cogagent: A visual language model for GUI agents. In: CVPR. pp. 14281–14290. IEEE (2024)
15. Hsiao, Y., Zubach, F., Baechler, G., Sunkara, S., Carbune, V., Lin, J., Wang, M., Zhu, Y., Chen, J.: Screenqa: Large-scale question-answer pairs over mobile app screenshots. In: NAACL (Long Papers). pp. 9427–9452. Association for Computational Linguistics (2025)
16. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. In: ICLR. OpenReview.net (2022)
17. Huang, Y., Wang, Y., Fu, Y.: D-code: Scaling image-pretrained vlms to video via dynamic compression and question decomposition. In: EMNLP. pp. 11798–11811. Association for Computational Linguistics (2025)
18. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: Layoutlmv3: Pre-training for document AI with unified text and image masking. In: ACM Multimedia. pp. 4083–4091. ACM (2022)
19. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., Madry, A., Baker-Whitcomb, A., Beutel, A., Borzunov, A., Carney, A., Chow, A., Kirillov, A., Nichol, A., Paino, A., Renzin, A., Passos, A.T., Kirillov, A., Christakis, A., Conneau, A., Kamali, A., Jabri, A., Moyer, A., Tam, A., Crookes, A., Tootoonchian, A., Kumar, A., Vallone, A., Karpathy, A., Braunstein, A., Cann, A., Codispoti, A., Galu, A., Kondrich, A., Tulloch, A., Mishchenko, A., Baek, A., Jiang, A., Pelisse, A., Woodford, A., Gosalia, A., Dhar, A., Pantuliano, A., Nayak, A., Oliver, A., Zoph, B., Ghorbani, B., Leimberger, B., Rossen, B., Sokolowsky, B., Wang, B., Zweig, B., Hoover, B., Samic, B., McGrew, B., Spero, B., Giertler, B., Cheng, B., Lightcap, B., Walkin, B., Quinn, B., Guarraci, B., Hsu, B., Kellogg, B., Eastman, B., Lugaresi, C., Wainwright, C.L., Bassin, C., Hudson, C., Chu, C., Nelson, C., Li, C., Shern, C.J., Conger, C., Barette, C., Voss, C., Ding, C., Lu, C., Zhang, C., Beaumont, C., Hallacy, C., Koch, C., Gibson, C., Kim, C., Choi, C., McLeavey, C., Hesse, C., Fischer, C., Winter, C., Czarnecki, C., Jarvis, C., Wei, C., Koumouzelis, C., Sherburn, D.: Gpt-4o system card. CoRR **abs/2410.21276** (2024)
20. Inoue, N., Kikuchi, K., Simo-Serra, E., Otani, M., Yamaguchi, K.: LayoutDM: Discrete Diffusion Model for Controllable Layout Generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10167–10176 (2023)

21. Jia, P., Li, C., Liu, Z., Shen, Y., Chen, X., Yuan, Y., Zheng, Y., Chen, D., Li, J., Xie, X., Zhang, S., Guo, B.: COLE: A hierarchical generation framework for graphic design. CoRR **abs/2311.16974** (2023)
22. Jin, C., Xu, H., Song, R., Lu, Z.: Text2poster: Laying out stylized texts on retrieved images. In: ICASSP. pp. 4823–4827. IEEE (2022)
23. Jyothi, A.A., Durand, T., He, J., Sigal, L., Mori, G.: Layoutvae: Stochastic scene layout generation from a label set. In: ICCV. pp. 9894–9903. IEEE (2019)
24. Kikuchi, K., Simo-Serra, E., Otani, M., Yamaguchi, K.: Constrained graphic layout generation via latent optimization. In: ACM Multimedia. pp. 88–96. ACM (2021)
25. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., Li, C.: Llava-onevision: Easy visual task transfer. Trans. Mach. Learn. Res. **2025** (2025)
26. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., Li, C.: Llava-onevision: Easy visual task transfer. Trans. Mach. Learn. Res. **2025** (2025)
27. Li, J., Yang, J., Hertzmann, A., Zhang, J., Xu, T.: Layoutgan: Generating graphic layouts with wireframe discriminators. In: ICLR (Poster). OpenReview.net (2019)
28. Li, J., Xu, Y., Lv, T., Cui, L., Zhang, C., Wei, F.: Dit: Self-supervised pre-training for document image transformer. In: ACM Multimedia. pp. 3530–3539. ACM (2022)
29. Li, J., Li, D., Savarese, S., Hoi, S.C.H.: BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML. Proceedings of Machine Learning Research, vol. 202, pp. 19730–19742. PMLR (2023)
30. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-1013/, accessed: 2026-06-27
31. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (January 2024), https://llava-vl.github.io/blog/2024-01-30-llava-next/, accessed: 2026-06-27
32. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS (2023)
33. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (Poster). OpenReview.net (2019)
34. Lu, J., Wang, H., Xu, Y., Wang, Y., Yang, K., Fu, Y.: Representation potentials of foundation models for multimodal alignment: A survey. In: EMNLP. pp. 16669–16684. Association for Computational Linguistics (2025)
35. Lu, J., Wang, H., Yang, K., Zhang, Y., Jenni, S., Fu, Y.: The indra representation hypothesis for multimodal alignment. In: NeurIPS (2025)
36. Malakouti, S., Aghazadeh, A., Khandelwal, A., Kovashka, A.: Benchmarking vlms' reasoning about persuasive atypical images. In: WACV. pp. 4788–4798. IEEE (2025)
37. Masry, A., Long, D.X., Tan, J.Q., Joty, S.R., Hoque, E.: Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In: ACL (Findings). pp. 2263–2279. Association for Computational Linguistics (2022)
38. Mathew, M., Bagal, V., Tito, R., Karatzas, D., Valveny, E., Jawahar, C.V.: Infographicvqa. In: WACV. pp. 2582–2591. IEEE (2022)
39. OpenAI: Gpt-5 system card. https://openai.com/index/introducing-gpt-5/ (2025), accessed: 2026-06-27
40. OpenAI: Gpt-o3 system card. https://openai.com/index/introducing-o3-and-o4-mini/ (2025), accessed: 2026-06-27

41. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: ACL. pp. 311–318. ACL (2002)
42. Patnaik, S., Jain, R., Krishnamurthy, B., Sarkar, M.: Aesthetiq: Enhancing graphic layout design via aesthetic-aware preference alignment of multi-modal large language models. In: CVPR. pp. 23701–23711. Computer Vision Foundation / IEEE (2025)
43. Pfitzmann, B., Auer, C., Dolfi, M., Nassar, A.S., Staar, P.W.J.: Doclaynet: A large human-annotated dataset for document-layout analysis. CoRR **abs/2206.01062** (2022)
44. Pu, Y., Zhao, Y., Tang, Z., Yin, R., Ye, H., Yuan, Y., Chen, D., Bao, J., Zhang, S., Wang, Y., Liang, L., Wang, L., Li, J., Li, X., Lian, Z., Huang, G., Guo, B.: ART: anonymous region transformer for variable multi-layer transparent image generation. In: CVPR. pp. 7952–7962. Computer Vision Foundation / IEEE (2025)
45. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (2021)
46. Si, C., Zhang, Y., Li, R., Yang, Z., Liu, R., Yang, D.: Design2code: Benchmarking multimodal code generation for automated front-end engineering. In: NAACL (Long Papers). pp. 3956–3974. Association for Computational Linguistics (2025)
47. Suzuki, T., Liu, K.J., Inoue, N., Yamaguchi, K.: Layerd: Decomposing raster graphic designs into layers. In: ICCV (2025)
48. Team, G.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. CoRR **abs/2507.06261** (2025)
49. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models. CoRR **abs/2302.13971** (2023)
50. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., Lin, J.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. CoRR **abs/2409.12191** (2024)
51. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., Wang, Z., Chen, Z., Zhang, H., Yang, G., Wang, H., Wei, Q., Yin, J., Li, W., Cui, E., Chen, G., Ding, Z., Tian, C., Wu, Z., Xie, J., Li, Z., Yang, B., Duan, Y., Wang, X., Hou, Z., Hao, H., Zhang, T., Li, S., Zhao, X., Duan, H., Deng, N., Fu, B., He, Y., Wang, Y., He, C., Shi, B., He, J., Xiong, Y., Lv, H., Wu, L., Shao, W., Zhang, K., Deng, H., Qi, B., Ge, J., Guo, Q., Zhang, W., Zhang, S., Cao, M., Lin, J., Tang, K., Gao, J., Huang, H., Gu, Y., Lyu, C., Tang, H., Wang, R., Lv, H., Ouyang, W., Wang, L., Dou, M., Zhu, X., Lu, T., Lin, D., Dai, J., Su, W., Zhou, B., Chen, K., Qiao, Y., Wang, W., Luo, G.: Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. CoRR **abs/2508.18265** (2025)
52. Xinru Zheng, Xiaotian Qiao, Y.C., Lau, R.W.: Content-aware generative modeling of graphic design layouts. ACM Transactions on Graphics (Proc. of SIGGRAPH 2019) **38** (2019)
53. Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florêncio, D.A.F., Zhang, C., Che, W., Zhang, M., Zhou, L.: Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In: ACL. pp. 2579–2591. Association for Computational Linguistics (2021)

54. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: Layoutlm: Pre-training of text and layout for document image understanding. In: KDD. pp. 1192–1200. ACM (2020)
55. Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., Chen, E.: A survey on multimodal large language models. CoRR **abs/2306.13549** (2023)
56. Zhang, H., Fu, Y.: Vqtoken: Neural discrete token representation learning for extreme token reduction in video large language models. Advances in Neural Information Processing Systems **38**, 32851–32869 (2026)
57. Zhang, H., Li, Y., He, S., Nagarajan, T., Chen, M., Lu, J., Li, A., Fu, Y.: Thinkjepa: Empowering latent world models with large vision-language reasoning model. CoRR **abs/2603.22281** (2026)
58. Zheng, L., Chiang, W., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E.P., Zhang, H., Gonzalez, J.E., Stoica, I.: Judging llm-as-a-judge with mt-bench and chatbot arena. In: NeurIPS (2023)
59. Zhong, X., Tang, J., Jimeno-Yepes, A.: Publaynet: Largest dataset ever for document layout analysis. In: ICDAR. pp. 1015–1022. IEEE (2019)