ONE DESIGN, MANY CANVASES: SYNTHESIZING LAYOUT PAIRS TO LEARN LAYERED DESIGN ADAPTATION

Anonymous authors
Paper under double-blind review

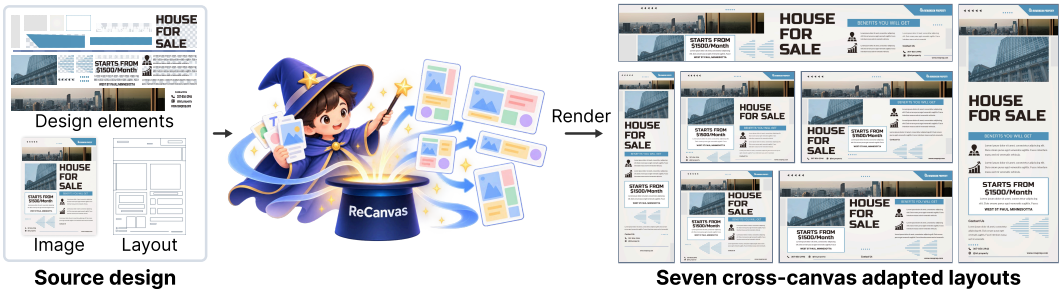

Figure 1: Cross-canvas layout adaptation with ReCanvas: given a source graphic design and its layered elements, ReCanvas adapts the layout to target canvases of different sizes and aspect ratios while preserving the source layout structure.

ABSTRACT

Adapting layered graphic designs to different canvases requires rearranging layered elements while preserving their grouping structure, reading order, and relative sizes. Beyond scaling and repositioning, different element types also require specific adaptations: text may reflow, while photos may be cropped. Learning such layout adaptation is challenging because paired layered designs containing the same elements across different canvas sizes are scarce. We address this limitation with a scalable layout-pair synthesis pipeline that, given a source design and target canvas size, generates an adapted target image and then recovers its layout using a trained VLM. Using this pipeline, we construct RECANVAS-DATA, a dataset of 4.22M source–target layout pairs derived from 292K layered designs spanning posters, presentation slides, and infographics. To learn source-to-target adaptation from RECANVAS-DATA, we develop ReCanvas, a vision-language model based on our proposed Layout-Structure Chain-of-Thought (LS-CoT) method. LS-CoT first infers a scene tree that captures the nested grouping and reading order of elements in the source design, and then conditions target-layout prediction on the inferred tree. Experiments show that ReCanvas outperforms baselines in layout quality while more faithfully preserving the source layout structure across canvas sizes.

1 INTRODUCTION

Graphic designs communicate information through design elements such as text, images, and graphics (Barnard, 2013). These elements are often organized into editable layers, allowing a design to be reused across different contexts and canvas sizes (Li et al., 2020). For example, presentation slides may need to fit different displays, while posters are often reformatted for different social media platforms. However, adapting a layered design to a new canvas is more than simply resizing the source design. It requires fitting the content to a new spatial constraint while preserving the structure and visual intent of the source. This often involves coordinated changes to both the layout and the elements themselves. At the layout level, elements need to be reorganized to fit the new canvas while preserving key structural properties of the source, such as grouping, reading order, and relative sizes. At the element level, different element types require different adjustments: text can be reflowed, photos can be cropped, and graphical elements such as underlays and charts can be resized and reshaped. We refer to this joint adaptation of the layout and elements as **layout adaptation** (Fig. 1).

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

Table 1: Comparison with existing graphic design datasets. # Cross-Canvas Pairs counts directed source–target pairs.

Dataset	# Layouts	Design Types			Design Elements			# Cross-Canvas Pairs
		Poster	Slide	Info.	# Text	# Non-Text	# Overflow	
PKU PosterLayout	9,974	✓	✗	✗	–	–	–	–
CGL	60,548	✓	✗	✗	–	–	–	–
Zenodo10K	264,899	✗	✓	✗	–	–	–	–
InfoDet	101,264	✗	✗	✓	–	–	–	–
GenPoster-100K	102,736	✓	✗	✗	601,290	–	–	–
Crello	23,302	✓	✓	✓	98,603	150,194	42,043	–
LICA†	1,550,244	✓	✓	✓	8,146,222	11,215,884	–	–
RECANVAS-DATA	1,188,318	✓	✓	✓	6,231,728	10,691,299	1,400,347	4,222,410

† Only a 1,183-layout subset of LICA is publicly available.

Existing methods train vision-language models (VLMs) to generate layouts from design elements and canvas specifications (Wei et al., 2025; Lin et al., 2025; Cheng et al., 2025). When adapting a design to a new canvas, these methods typically generate the target layout from the elements and target canvas size alone, without using the source layout (Wei et al., 2025). In effect, they treat adaptation as layout generation from scratch, requiring the model to reconstruct the composition from individual elements rather than transform the source layout. This makes it difficult to preserve the source design's visual organization and intent. Paired layouts of the same design across different canvas sizes could provide direct supervision for learning such source-conditioned adaptation. However, such pairs are scarce in existing layered graphic design datasets (Yamaguchi, 2021; Hsu et al., 2023; Wang et al., 2025; Hirsch et al., 2026), and manually creating them is also costly because each target layout requires careful adjustment by a designer.

To bridge this data gap, we introduce a scalable pipeline for synthesizing cross-canvas layout pairs. Given a layered source design, the pipeline first uses an image editing model, such as GPT-Image-2 (OpenAI, 2026b), to adapt it to a target canvas in pixel space, allowing coordinated changes such as element repositioning, resizing, text reflow, and image cropping. It then grounds the source elements in the generated image to recover an editable target layout, including element-specific changes such as crop regions and text formatting. We apply this pipeline to 292K layered designs spanning posters, presentation slides, and infographics, collected from existing datasets (Yamaguchi, 2021; Wang et al., 2025; Li et al., 2026) and online resources (SlidesCarnival, 2026; PPTMON, 2026). This resulting dataset, **RECANVAS-DATA**, includes 1.19M layered designs and 4.22M directed layout pairs (Table 1) for training layout adaptation. Building on RECANVAS-DATA, we develop **ReCanvas**, a vision-language model that predicts the target layout from the source design and the target canvas size. Its key component is **Layout-Structure Chain-of-Thought (LS-CoT)**, which first infers a scene tree from the source design and then conditions target-layout prediction on this tree. The scene tree organizes elements into nested groups and orders sibling nodes by their intended reading order, providing explicit intermediate supervision to preserve source layout structure.

We evaluate ReCanvas on **RECANVAS-BENCH**, a 1,500-task benchmark spanning a wide range of element numbers and aspect-ratio changes, and compare it with general-purpose VLMs and specialized layout-generation models. We assess the adapted layouts using VLM-based evaluation, programmatic metrics for readability and source-layout similarity, and human evaluation. Across these evaluations, ReCanvas consistently outperforms the baselines, showing better preservation of the source design and higher adaptation quality across target canvases. To facilitate future research on layered graphic design adaptation, we will publicly release RECANVAS-DATA upon publication.

The key contributions of this work are as follows:

- We introduce a scalable layout-pair synthesis pipeline that adapts layered designs in pixel space and recovers editable layouts, enabling large-scale construction of cross-canvas layout pairs.
- We construct RECANVAS-DATA with 1.19M layered designs and 4.22M source–target layout pairs, and introduce RECANVAS-BENCH, a 1,500-task benchmark for layout adaptation.
- We develop ReCanvas, a vision-language model for cross-canvas layered design adaptation that uses the proposed LS-CoT method to explicitly reason about source layout structure.

2 Related Work

Our work bridges layered graphic design datasets and graphic design layout generation.

Layered graphic design datasets Existing layout datasets provide either layout annotations, such as PosterLayout (Hsu et al., 2023) and CGL-Dataset (Zhou et al., 2022; Li et al., 2023), or layered representations with editable elements, such as Crello (Yamaguchi, 2021), GenPoster-100K (Wang et al., 2025), and LICA (Hirsch et al., 2026). While these datasets provide supervision for learning individual layouts, cross-canvas layout adaptation requires paired layouts showing how the same design changes across different canvas sizes. Such editable pairs are rarely available, and creating them manually is costly because each target design requires careful adjustment. We address this gap by first adapting layered designs in pixel space with an image editing model and then grounding the source elements in the generated images to recover editable target layouts. This pipeline enables large-scale synthesis of cross-canvas layout pairs.

Graphic design layout generation Early layout generation methods model layouts as sets or sequences of labeled bounding boxes and generate them using GANs (Li et al., 2019), VAEs (Jyothi et al., 2019; Lee et al., 2020), autoregressive transformers (Gupta et al., 2021; Kong et al., 2022; Jiang et al., 2023; Horita et al., 2024), and diffusion models (Inoue et al., 2023; Zhang et al., 2023; Levi et al., 2023; Chen et al., 2024). LLM-based methods later formulate layout generation as structured language generation (Feng et al., 2023; Lin et al., 2023; Tang et al., 2024). More recent VLM-based methods further incorporate layered design elements as input, enabling graphic layout generation conditioned on element appearance (Zhu et al., 2024; Cheng et al., 2025; Lin et al., 2025; Wei et al., 2025; Guo et al., 2026). These methods generally generate layouts without conditioning on an existing source layout. Our work instead studies cross-canvas layout adaptation, where the target layout should preserve the layout structure of the source design.

ReLayout (Lin et al., 2026) is more closely related, as it conditions generation on pairwise size and position relationships extracted from the source layout. However, these pairwise relationships describe local geometry and may change substantially when adapting a design to a different canvas. In contrast, our method moves beyond local pairwise geometry by conditioning on the full source design and representing its higher-level structure with an intermediate scene tree, which captures organization such as grouping and reading order that should be preserved across canvases.

3 ReCanvas-Data Construction

Cross-canvas layout adaptation requires paired examples showing how the same layered design changes across canvases. To provide such supervision at scale, we construct ReCanvas-Data, a layered graphic design dataset with paired layouts across multiple canvas sizes. As shown in Fig. 2, our construction pipeline has four stages. We first process three types of graphic designs, including posters, presentation slides, and infographics, collected from five sources. We then use an image editing model to adapt each design to multiple target canvases in pixel space. Next, we train a VLM to ground the source elements in the generated images and recover structured target layouts. Finally, we filter low-quality target layouts and form source–target pairs from valid layout variants of the same design across different canvases.

Source design processing We collect layered graphic designs spanning three design types with distinct composition patterns. Posters are drawn from Crello (Yamaguchi, 2021) and GenPoster-100K (Wang et al., 2025), presentation slides are collected from SlidesCarnival (SlidesCarnival, 2026) and PPTMON (PPTMON, 2026), and infographics are generated using the ChartGalaxy pipeline (Li et al., 2026). Details of the data sources and their usage policies are provided in Appendix B.1. The ethical considerations in the data collection process are further discussed in Ethics Statement.

The collected designs are stored in heterogeneous formats, including PSD, PPTX, and SVG, and therefore require preprocessing into a unified layered representation. We extract image and text elements together with their geometry and type-specific attributes. For elements extending beyond the canvas, we retain their full uncropped content rather than only the visible region, allowing hidden content to become visible after adaptation. For text elements, we preserve the text content and available typographic attributes to support subsequent text reflow and re-rendering. We further remove invisible elements, consolidate highly fragmented layers, merge effect layers into the background,

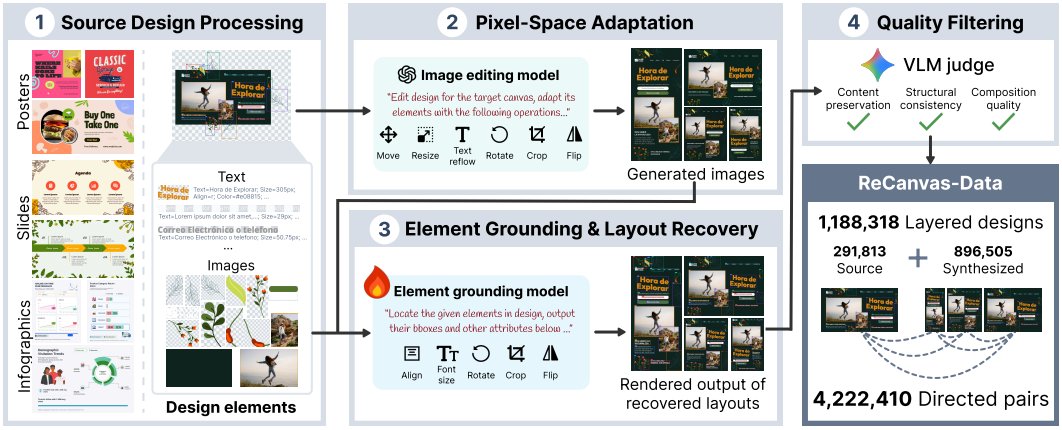

Figure 2: Overview of the RECANVAS-DATA construction pipeline.

and discard designs with more than 80 elements. Near-duplicate designs are removed using CLIP similarity (Radford et al., 2021) and Perceptual Hashing. After preprocessing and deduplication, we hold out a subset of 6,128 designs for evaluation and retain 291,813 layered source designs with 3,964,831 editable elements for training and subsequent data synthesis.

Pixel-space adaptation Directly synthesizing editable target layouts requires coordinating both layout-level changes, such as repositioning and resizing, and element-level changes, such as text reflow and image cropping. We therefore first adapt each source design in pixel space using GPT-Image-2 (OpenAI, 2026b), instructing it to fit the rendered source design to a target canvas while preserving its content and overall structure. This allows diverse changes, including text reflow, image cropping, rotation, and flipping, to be handled jointly. To obtain diverse canvas changes, we select target canvases through balanced sampling over design-type-specific aspect-ratio sets, with constraints that avoid near-duplicate canvas shapes. This process produces 1,082,848 target renderings before quality filtering. The complete candidate sets and sampling procedure are provided in Appendix B.2.

Element grounding and layout recovery Although pixel-space adaptation produces visually complete target designs, the generated images are flattened and therefore cannot directly provide editable layout supervision. Motivated by recent advances in visual grounding and object detection (Chen et al., 2023; Zhu et al., 2026; Wang et al., 2026), we train an element grounding model to locate the source design elements in each generated target image and recover their target layout attributes. Given the generated image $\widetilde{I}_t$ and the source elements $\mathcal{E}$, the model predicts

$$g_{\text{rec}}(\widetilde{I}_t, \mathcal{E}) = \{(b_i, z_i, \theta_i, \mathbf{a}_i)\}_{i=1}^{N}, \tag{1}$$

where b_i, z_i, and θ_i denote the bounding box, stacking order, and rotation of element e_i, respectively. The type-specific attributes $\mathbf{a}_i$ include crop region and horizontal flip for image elements, and font size and alignment for text elements. We train the grounding model on the processed layered designs together with an internal dataset of 5.63M layered designs, using rendered images and known element attributes as direct supervision. Image elements are provided with their full uncropped content, enabling the model to infer the visible crop region from the rendered design. For text elements, the model receives the text content and predicts typographic attributes from its appearance. To improve recovery of less frequent geometric transformations, we additionally apply rotation and horizontal-flip augmentation to 2% of image elements during training. We implement the grounding model with Qwen3.5-2B (Qwen Team, 2026).

Quality filtering and pair construction Generated images and recovered layouts may contain editing or grounding errors. We therefore apply a two-stage quality filtering procedure. First, we remove structurally invalid results using programmatic checks, including those with missing or duplicated elements, fully invisible elements, and severe text occlusion. For each remaining image, we re-render the recovered layout using the source elements and perform a perceptual quality assessment with Gemini-3.6-Flash (Google, 2026). The evaluator compares the source and re-rendered target designs and assesses whether the target forms a valid adaptation in terms of content preservation, structural consistency, and overall composition quality. Only samples passing both stages are retained. To validate the reliability of the VLM-based assessment, we recruit a designer to manually inspect

1,000 randomly sampled decisions and observe 98.9% agreement. This process retains 896,505 generated target variants (82.8%), corresponding to an average of 3.07 accepted variants per source design. After filtering, each design is associated with its original layout and multiple valid layout variants across different canvases. Any two variants of the same design can serve bidirectionally as source and target, forming 4,222,410 directed cross-canvas adaptation pairs.

Dataset statistics Table 1 compares RECANVAS-DATA with existing public layered graphic design datasets. Unlike prior datasets, which primarily provide individual layered designs, RECANVAS-DATA includes paired layouts of the same design across different canvases. The source designs contain a median of 12 elements, with 11.7% containing more than 20. Across the layout pairs, the median source–target aspect-ratio change is $1.88\times$, with 44.9% of pairs exceeding $2\times$ and 18.2% exceeding $3\times$. Detailed distributions and examples are provided in Appendix B.2 and B.3.

4 RECANVAS MODEL

Building on RECANVAS-DATA, we develop the ReCanvas model for source-conditioned cross-canvas layout adaptation. Below, we describe the model overview, the LS-CoT method, and the training strategy, with implementation details in Appendix D.

4.1 MODEL OVERVIEW

We build ReCanvas on Qwen3.5-9B (Qwen Team, 2026) and extend its visual encoder to accept RGBA inputs. This allows the model to directly process image elements represented as RGBA layers (Cheng et al., 2025), where the alpha channel preserves transparency and provides cues about element shape and visible regions. Since the original visual encoder accepts only RGB images, we expand its visual patch embedding from three to four input channels while keeping the remaining vision and language architecture unchanged. We refer to the resulting model as Qwen3.5-9B-α.

Given a source design and a target canvas, ReCanvas receives layered design elements, the rendered source image, the source layout, and the target canvas specification as input (Fig. 3). Image elements are provided as RGBA layers, while text elements retain their textual content and typographic attributes. The model autoregressively predicts a scene tree that captures the source layout structure, followed by the adapted target layout and its element-specific attributes.

4.2 THE LS-COT METHOD

Cross-canvas adaptation often involves substantial geometric changes, while higher-level design structures such as semantic grouping and reading order should remain consistent. Directly predicting target coordinates provides no explicit mechanism for reasoning about such structures. We therefore introduce the LS-CoT method, which guides the model to first infer the structural organization of the source design before predicting its target layout.

Let $X = (\mathcal{E}, \mathcal{L}_s, I_s, c_t)$ denote the model input, consisting of the source elements $\mathcal{E}$, source layout $\mathcal{L}_s$, rendered source image I_s, and target canvas c_t, and let $\mathcal{L}_t$ denote the target layout. Given X, LS-CoT first infers a source scene tree $\mathcal{T}_s$ (Fig. 3). Each leaf of $\mathcal{T}_s$ corresponds to exactly one

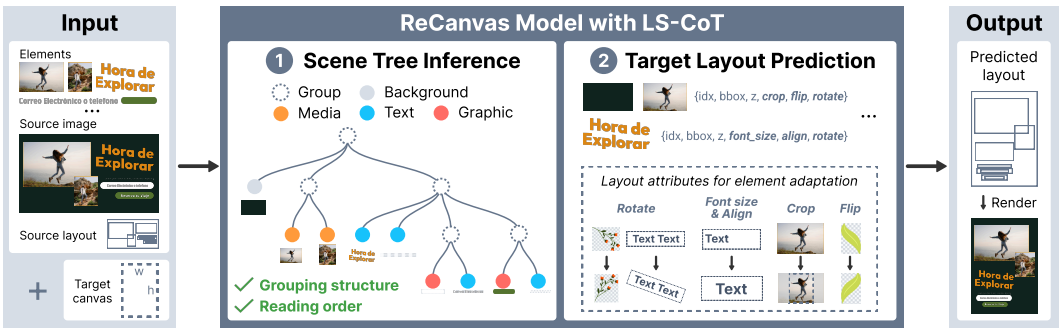

Figure 3: An overview of our ReCanvas model with the LS-CoT method.

element e_i, while internal nodes represent nested semantic groups. Sibling nodes follow the reading or design order of the source layout. Each leaf also carries an auxiliary semantic category drawn from a hierarchical taxonomy, with coarse categories including background, text, media, and graphic, and finer-grained subcategories such as title, body text, hero image, and data visualization.

Conventional layout generation methods directly model $p_\theta(\mathcal{L}_t \mid X)$, where p_θ denotes the autoregressive distribution parameterized by the model, requiring the source layout structure to be captured implicitly from the input. In contrast, LS-CoT introduces $\mathcal{T}_s$ as an explicit intermediate representation and factorizes the prediction as

$$p_\theta(\mathcal{T}_s, \mathcal{L}_t \mid X) = p_\theta(\mathcal{T}_s \mid X)\, p_\theta(\mathcal{L}_t \mid X, \mathcal{T}_s). \tag{2}$$

The model therefore first generates the source scene tree and then predicts the target layout within the same autoregressive sequence. This makes the source grouping and reading-order structure explicit before target-layout prediction, rather than requiring the model to infer this structure implicitly while generating target coordinates. To obtain scene-tree supervision at scale, we use Gemini-3.6-Flash (Google, 2026) to annotate source designs, prompting it to construct trees in which each element appears exactly once as a leaf, with hierarchical grouping, ordered siblings, and semantic labels for leaf elements. To verify annotation quality, we recruit two designers to manually inspect 500 randomly sampled generated trees for grouping, sibling-ordering, and coverage errors, with 97.2% judged structurally correct. Further annotation details are provided in Appendix E.

4.3 Training Strategy

We train ReCanvas with a three-stage curriculum. The first stage learns general layout generation. The second stage specializes the model for source-conditioned cross-canvas adaptation with LS-CoT. The final stage further improves generated layout quality with reinforcement learning.

Layout-generation SFT (LG-SFT) We first train the model on an internal layered design dataset containing 5.63M different layered designs to establish general layout-generation capability. We optimize the standard autoregressive negative log-likelihood objective.

Layout-adaptation SFT (LA-SFT) In the second stage, we fine-tune the model on the paired layouts in ReCanvas-Data. Each training example conditions on the target canvas and the source design, with the output following the LS-CoT format introduced in Fig. 3. We optimize the scene-tree and target-layout tokens jointly using the same autoregressive objective. This stage introduces direct supervision for source-to-target adaptation, enabling the model to learn how a specific source design should adapt across canvases.

Reinforcement learning (RL) Layout adaptation admits multiple plausible outputs (Pan & Liu, 2026), whereas supervised training associates each input with a single synthesized target layout. Optimizing toward this single target may therefore penalize other valid adaptations and over-constrain the model to one particular solution. Therefore, we further optimize ReCanvas with reinforcement learning using verifiable and VLM-based rewards.

For each input x_i, we sample a group of G rollouts $\{y_{i,j}\}_{j=1}^{G}$, where $y_{i,j} = [\widehat{\mathcal{T}}_{s,i,j}; \widehat{\mathcal{L}}_{t,i,j}]$ contains the predicted source scene tree and target layout. Before computing rewards, we apply a strict validity check to ensure that both outputs are parseable and contain the complete set of elements. Invalid rollouts are excluded from reward normalization and assigned a negative advantage.

For each valid rollout, we compute four verifiable rewards and one VLM-based reward:

$$\mathbf{r}_{i,j} = [R_{\text{tree}}, R_{\text{ele}}, R_{\text{text}}, R_{\text{rel}}, R_{\text{qual}}]_{i,j}\,. \tag{3}$$

$R_{\text{tree}} = \text{Sim}_{\text{tree}}(\widehat{\mathcal{T}}_s, \mathcal{T}_s)$ measures the consistency of the inferred scene tree with the source tree in semantic categories, hierarchical grouping, and element order. For the final layout, R_{ele} and R_{text} assess element visibility and text readability, while $R_{\text{rel}} = \text{Sim}_{\text{layout}}(\widehat{\mathcal{L}}_t, \mathcal{L}_s)$ measures the preservation of source-derived layout relationships. Finally, R_{qual} uses a VLM judge to provide perceptual feedback on the rendered adaptation. Thus, the reward signals provide a holistic assessment of the model output, covering both intermediate reasoning and the quality of the resulting layout.

Because these rewards differ in scale and variance, we adopt Group reward-Decoupled Normalization Policy Optimization (Liu et al., 2026). Let $r_{i,j}^{(k)}$ denote the k-th reward signal for rollout $y_{i,j}$.

We independently normalize each reward dimension within the rollout group as $\hat{r}_{i,j}^{(k)} = (r_{i,j}^{(k)} - \mu_i^{(k)})/(\sigma_i^{(k)} + \epsilon)$, where $\mu_i^{(k)}$ and $\sigma_i^{(k)}$ denote the mean and standard deviation of reward k over valid rollouts for input x_i. The normalized rewards are then weighted and aggregated as $s_{i,j} = \sum_k w_k \hat{r}_{i,j}^{(k)}$, followed by batch-level normalization to obtain the sequence-level advantage. We optimize the resulting advantages using a GRPO-style clipped objective (Schulman et al., 2017; Shao et al., 2024) with KL regularization. Appendix D.3 provides more details.

5 Experiments

5.1 Experimental Setup

Evaluation dataset We construct ReCanvas-Bench by sampling 500 designs from the 6,128-design held-out set, with 100 designs from each of the five data sources. The selected designs contains 4–50 elements. For each design, we choose three target canvas sizes corresponding to small, medium, and large aspect-ratio changes relative to the source, resulting in 1,500 adaptation tasks in total. We categorize aspect-ratio changes into three levels: small (1–1.5], medium (1.5–2.5], and large (>2.5). Additional benchmark statistics are provided in Appendix C.1.

Evaluation metrics We evaluate adapted designs using complementary perceptual and programmatic metrics (Zhou et al., 2022; Hsu et al., 2023; Yang et al., 2025; Li et al., 2026). For perceptual evaluation, we use Gemini-3.6-Flash (Google, 2026) to jointly assess the source and the rendered target design. This evaluator assigns a score from 0 to 100 based on element fidelity, text readability, structural preservation, and overall composition quality. We also use programmatic metrics to assess layout readability and source similarity. Layout readability measures whether elements remain sufficiently visible within the target canvas and text remains readable. Source similarity measures how well the adapted layout preserves key characteristics of the source layout, including alignment, relative size, occlusion, containment, and space utilization. In addition to the individual metrics, we compute an aggregated readability score as the percentage of outputs that satisfy all readability criteria, and an aggregated similarity score as the average of the individual similarity metrics. Outputs that cannot be parsed or do not contain exactly the same set of elements as the input are treated as invalid and receive a score of zero for all metrics. Further details are provided in Appendix C.2.

Baselines We compare ReCanvas with both general-purpose VLMs and specialized layout models. We evaluate 10 open-source and 2 proprietary general-purpose VLMs (Table 2), together with 3 layout-generation methods (Table 3): PosterCopilot (Wei et al., 2025), LaDeCo (Lin et al., 2025), and ReLayout (Lin et al., 2026).

Table 2: Main results on ReCanvas-Bench, comparing readability, source-layout similarity, and VLM score of the adapted layout.

Model	Params	Readability			Similarity						VLM Score				
		Elem.	Text	All	Align.	Size	Occl.	Cont.	Space Util.	Avg.	Elem. Fid.	Text Read.	Struct. Pres.	Comp. Qual.	Total
Full Score	–	100	100	100	100	100	100	100	100	100	20	20	30	30	100
Open-Source General-Purpose Models															
Kimi-VL-A3B	16B-A3B	37.09	11.39	4.20	84.22	16.99	31.31	26.81	41.26	40.12	1.53	1.56	1.61	1.50	6.20
Molmo2-8B	8B	28.82	20.68	10.13	34.56	23.90	25.71	20.45	20.57	25.04	2.73	2.98	3.17	2.75	11.62
Qwen2.5-VL-7B	7B	48.62	37.57	19.27	51.82	39.42	42.44	38.00	34.83	41.30	4.89	5.28	5.68	4.71	20.56
Qwen3.5-4B	4B	64.21	52.28	25.40	63.57	53.07	54.22	42.91	49.83	52.72	6.26	6.76	7.20	6.76	26.98
Qwen3.5-9B	9B	45.70	40.64	27.00	45.41	41.21	43.12	36.31	35.30	40.27	6.42	6.70	7.76	6.21	27.09
InternVL3.5-30B-A3B	30B-A3B	75.85	61.26	32.67	75.25	60.60	62.51	49.09	51.13	59.71	7.23	7.79	7.86	6.70	29.59
Gemma-4-12B	12B	90.23	66.86	23.40	90.07	68.44	73.47	62.92	67.95	72.57	7.71	7.78	8.91	9.41	33.82
Llama-4-Scout	109B-A17B	92.14	71.08	30.13	91.64	72.98	74.27	62.95	73.54	75.08	8.06	8.48	9.11	8.67	34.31
Gemma-4-26B-A4B	26B-A4B	74.77	64.57	38.07	74.29	63.11	67.56	50.07	66.61	64.33	8.70	9.23	11.03	10.58	39.54
Qwen3.5-27B	27B	97.34	89.94	57.67	96.12	79.60	89.21	90.52	81.12	87.31	13.10	13.64	16.19	14.63	57.56
Proprietary General-Purpose Models															
Gemini-3.6-Flash (High)	–	98.14	95.11	78.27	96.85	89.17	94.93	96.61	86.47	92.80	17.17	17.46	24.12	22.73	81.48
GPT-5.6-Sol (High)	–	98.79	97.04	82.87	97.26	**89.94**	**95.88**	**97.56**	79.39	92.01	18.45	18.67	25.78	23.95	86.85
Specialized Layout-Adaptation Model															
ReCanvas (Ours)	9B	**99.50**	**98.41**	**85.40**	**97.38**	87.62	94.88	96.63	**87.54**	**92.81**	**18.75**	**19.08**	**26.58**	**25.44**	**89.86**

Table 3: Results of layout adaptation under different training formulations and condition settings.

Method	Params	Source Condition	Readability			Similarity						VLM Score				
			Elem.	Text	All	Align.	Size	Occl.	Cont.	Space Util.	Avg.	Elem. Fid.	Text Read.	Struct. Pres.	Comp. Qual.	Total
Full Score	–	–	100	100	100	100	100	100	100	100	100	20	20	30	30	100
Layout-Generation Training (w/o Layout-Pair Supervision)																
PosterCopilot†	7B	None	38.02	29.74	14.67	37.38	27.86	20.34	6.60	26.53	23.74	2.93	3.23	2.91	2.91	11.99
LaDeCo†	8B	None	87.00	47.57	10.07	87.05	59.70	41.86	24.33	63.84	55.35	4.53	4.40	4.08	4.83	17.85
Qwen3.5-9B-α + LG-SFT	9B	None	92.01	82.11	49.60	90.45	71.59	73.90	48.73	74.18	71.77	12.80	13.19	15.83	16.00	57.82
		Pairwise Relations	97.46	91.92	64.67	95.36	81.09	91.02	88.48	82.22	87.63	15.09	15.49	19.88	19.29	69.76
Layout-Adaptation Training (w/ Layout-Pair Supervision)																
Qwen3.5-9B-α + LA-SFT	9B	Image	91.82	86.56	61.07	90.16	76.42	79.56	58.64	79.43	76.84	14.98	15.41	20.25	19.66	70.29
		Layout	96.83	93.29	70.20	95.06	83.56	91.30	90.64	84.82	89.08	16.61	17.07	23.00	22.04	78.72
		Image + Layout	**97.48**	**94.33**	**73.13**	**95.64**	**84.82**	**92.67**	**93.30**	**85.61**	**90.41**	**17.40**	**17.75**	**24.23**	**23.19**	**82.58**

† Official checkpoints are evaluated directly on RECANVAS-BENCH without additional fine-tuning.

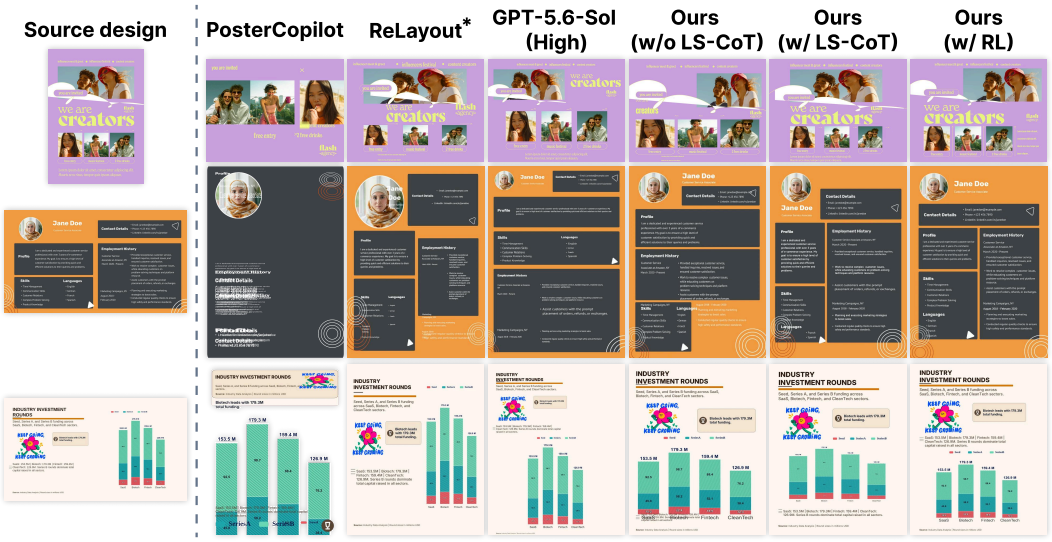

Figure 4: Qualitative comparison of our model with baselines for layout adaptation. * ReLayout is reproduced with Qwen3.5-9B-α and trained on RECANVAS-DATA without layout-pair supervision.

5.2 MAIN RESULTS

Comparison with general-purpose VLMs Table 2 reports the results on RECANVAS-BENCH across both open-source and proprietary general-purpose models. ReCanvas achieves higher overall readability, similarity, and VLM-based evaluation scores than all compared models, including GPT-5.6-Sol (OpenAI, 2026a) and Gemini-3.6-Flash (Google, 2026) under high reasoning effort. We further examine the effect of reasoning budget on proprietary VLMs. As shown in Table 4, increasing the reasoning effort improves the VLM score from 69.84 to 81.48 for Gemini-3.6-Flash and from 81.54 to 86.85 for GPT-5.6-Sol, at the cost of much longer reasoning sequences. In comparison, our current LS-CoT uses only 225.5 reasoning tokens on average. Future work can explore enabling more extensive reasoning for layout adaptation.

Table 4: Effect of reasoning level on average reasoning length and overall VLM score.

Model	Level	Length	VLM Score
Gemini-3.6-Flash	Low	3.3	69.84
	High	13,455.7	81.48 (+11.64)
GPT-5.6-Sol	Low	878.0	81.54
	High	4,876.4	86.85 (+5.31)
ReCanvas (Ours)	–	225.5	**89.86**

Effect of paired supervision and source conditioning To examine the effect of paired supervision, we compare models trained with and without source–target layout pairs (Table 3). We first evaluate PosterCopilot (Wei et al., 2025) and LaDeCo (Lin et al., 2025) using their official checkpoints without additional fine-tuning. As their original training distributions differ from RECANVAS-BENCH, particularly in the number of design elements, these results mainly reflect their out-of-the-box generalization to our task. For a controlled comparison, we further train two Qwen3.5-9B-α baselines on the individual layouts in RECANVAS-DATA without using layout-pair supervision. The source-

Table 5: Incremental ablation of model design choices across aspect-ratio changes and element-count ranges. Values are overall VLM scores, with changes from the preceding setting shown in parentheses.

Setting	Aspect-Ratio Change			Element Number			Overall
	Small	Medium	Large	1–10	11–25	> 25	
Qwen-3.5-9B + LA-SFT	89.43	75.89	72.04	83.97	82.38	74.19	79.12
+ Alpha Channel	88.46 (-0.97)	79.87 (+3.98)	79.40 (+7.36)	86.56 (+2.59)	85.39 (+3.01)	78.41 (+4.22)	82.58 (+3.46)
+ LG-SFT	89.44 (+0.98)	81.03 (+1.16)	80.58 (+1.18)	86.52 (-0.04)	86.98 (+1.59)	79.63 (+1.22)	83.68 (+1.10)
+ LS-CoT	91.05 (+1.61)	85.89 (+4.86)	84.98 (+4.40)	87.79 (+1.27)	89.18 (+2.20)	85.50 (+5.87)	87.31 (+3.63)
+ RL	**93.00** (+1.95)	**88.15** (+2.26)	**88.42** (+3.44)	**90.34** (+2.55)	**90.98** (+1.80)	**88.69** (+3.19)	**89.86** (+2.55)

free baseline conditions only on the design elements and target canvas, while the ReLayout-style baseline (Lin et al., 2026) additionally conditions on pairwise size and position relations derived from the source layout. Source-free training yields limited adaptation performance, whereas adding source relations substantially improves the average similarity score to 87.63. However, its overall VLM score remains only 69.76, below models trained with paired adaptation supervision. As shown in Fig. 4, preserving source relations too rigidly can produce layouts that remain similar to the source but do not fit the target canvas well. In contrast, training on source–target layout pairs enables the model to learn which source characteristics should be preserved and which should change across canvases. Within this setting, conditioning on the source layout is more effective than using the source image alone, while combining both achieves the best performance, with an overall VLM score of 82.58. Additional qualitative results of ReCanvas are provided in Appendix A.

Ablation of model design choices Table 5 shows the effects of progressively adding each model component across different adaptation tasks. Starting from the Qwen3.5-9B baseline, adding the alpha channel improves the overall VLM score from 79.12 to 82.58. Layout-generation SFT further improves the score to 83.68. Adding LS-CoT yields the largest gain among the training components, increasing the overall score to 87.31. The improvement is particularly pronounced for designs with more than 25 elements (+5.87) and for medium and large aspect-ratio changes (+4.86 and +4.40, respectively), suggesting that LS-CoT is especially beneficial for more complex designs and larger canvas changes. As shown in Fig. 4, explicitly modeling the source layout structure helps preserve coherent organization in more challenging cases. Finally, reinforcement learning further improves performance across all groups, raising the overall score to 89.86. The last column of Fig. 4 also shows that reinforcement learning produces more visually harmonious layouts.

Human evaluation We further evaluate the models with human judgments and examine the alignment between human preferences and our VLM-based evaluation. For each of the 500 source designs in RECANVAS-BENCH, we select one target canvas and sample a pair of valid outputs from the seven top-performing models (Table 6). We balance model pairs and aspect-ratio change levels, and recruit three designers to independently provide blinded pairwise preferences, yielding 1,500 judgments in total. ReCanvas achieves a human win rate of 83.90%, compared with 67.20% for GPT-5.6-Sol (High) and 66.90% for Gemini-3.6-Flash (High), and achieves the highest Bradley–Terry worth of 50.38. Overall, the human and VLM evaluations produce the same ordering across the models. Further agreement analysis shows substantial consistency between the human and VLM evaluators. Additional details are provided in Appendix F.

Table 6: Results of pairwise comparison of seven models under human and Gemini-3.6-Flash evaluations. BT worth is estimated using the Davidson extension and normalized to sum to 100 within each evaluator.

Model	Human Eval		Gemini-3.6-Flash Eval	
	Win Rate	BT Worth	Win Rate	BT Worth
ReCanvas(Ours)	**83.90**	**50.38**	**78.00**	**36.59**
GPT-5.6-Sol (High)	67.20	17.87	69.60	23.41
Gemini-3.6-Flash (High)	66.90	17.45	65.40	18.93
GPT-5.6-Sol (Low)	50.60	7.37	53.50	10.83
Gemini-3.6-Flash (Low)	40.80	4.32	42.00	6.20
Qwen3.5-27B	23.40	1.58	22.50	2.24
Gemma-4-26B-A4B	17.00	1.03	18.90	1.80

6 CONCLUSION

In this paper, we introduce a scalable pipeline for synthesizing cross-canvas layout pairs and use it to construct RECANVAS-DATA, a large-scale dataset for cross-canvas layout adaptation. Building on this dataset, we develop ReCanvas, which uses the proposed LS-CoT method to explicitly model source layout structure during adaptation. Experiments on RECANVAS-BENCH demonstrate consistent improvements over general-purpose VLMs and specialized layout-generation models across automatic and human evaluations. These results demonstrate the benefits of explicitly modeling source layout structure for cross-canvas layout adaptation.

AI USE STATEMENT

Generative AI tools were used as components of our research methodology, including synthetic data generation and quality assessment, as described in the corresponding sections of the paper. We additionally used generative AI tools to assist with code implementation and debugging, and to improve the clarity and readability of the manuscript. The illustration in the teaser figure was generated with the assistance of a generative image model. All AI-assisted code and manuscript content were reviewed and verified by the authors, and the authors take full responsibility for the methodology, results, figures, and final content of the paper.

ETHICS STATEMENT

Dataset construction Our work uses layered graphic designs obtained from publicly available datasets and online template resources. We document the provenance, licenses, and applicable usage terms of all data sources in Appendix B.1. These sources are used in accordance with their stated licenses and source-specific terms, including attribution and non-commercial restrictions where applicable. Because some sources impose additional restrictions on redistribution or contain third-party assets under separate licenses, we do not assume that a source-level license grants unrestricted redistribution of all underlying content. Our release will therefore include only data and derived artifacts permitted by the corresponding source terms. For materials that cannot be redistributed, we provide source references and processing code rather than the layered design elements.

The dataset is intended for research on graphic design layout adaptation. We do not collect private user designs or personal data specifically for this study. Nevertheless, publicly available graphic designs may contain incidental depictions of people, names, or other content inherited from their original sources, and may reflect cultural or demographic biases present in those sources. We therefore preserve source provenance and encourage downstream users to consider these limitations when applying the dataset beyond the research settings studied in this work.

Human evaluation Our human evaluation involved three designers who provided blinded pairwise preferences between 500 pairs of generated graphic designs. Participants were informed of the study procedure, provided consent before participation, and were compensated 30$ for their time. The evaluation involved only non-sensitive graphic design content, with no violent, sexual, or otherwise disturbing material, and no personally identifiable information was collected beyond what was necessary for study administration. We consider the evaluation to pose minimal risk to participants.

REFERENCES

Malcolm Barnard. *Graphic design as communication*. Routledge, 2013.

Changjian Chen, Yukai Guo, Fengyuan Tian, Shilong Liu, Weikai Yang, Zhaowei Wang, Jing Wu, Hang Su, Hanspeter Pfister, and Shixia Liu. A unified interactive model evaluation for classification, object detection, and instance segmentation in computer vision. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):76–86, 2023.

Jian Chen, Ruiyi Zhang, Yufan Zhou, Rajiv Jain, Zhiqiang Xu, Ryan Rossi, and Changyou Chen. Towards aligned layout generation via diffusion model with aesthetic constraints. In *Proceedings of the International Conference on Learning Representations*, 2024.

Yutao Cheng, Zhao Zhang, Maoke Yang, Hui Nie, Chunyuan Li, Xinglong Wu, and Jie Shao. Graphic design with large multimodal model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2473–2481, 2025.

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. In *Proceedings of the Conference on Neural Information Processing Systems*, pp. 18225–18250, 2023.

Google. Gemini 3.6 flash, 2026. URL `https://deepmind.google/models/model-cards/gemini-3-6-flash/`. Accessed: 2026-9-1.

Junrong Guo, Shancheng Fang, Yadong Qu, and Hongtao Xie. Seeing is improving: Visual feedback for iterative text layout refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25893–25903, 2026.

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

Elad Hirsch, Shubham Yadav, Mohit Garg, and Purvanshi Mehta. LICA: Layered image composition annotations for graphic design research. *arXiv preprint arXiv:2603.16098*, 2026.

Daichi Horita, Naoto Inoue, Kotaro Kikuchi, Kota Yamaguchi, and Kiyoharu Aizawa. Retrieval-augmented layout transformer for content-aware layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 67–76, 2024.

Hsiao Yuan Hsu, Xiangteng He, Yuxin Peng, Hao Kong, and Qing Zhang. PosterLayout: A new benchmark and approach for content-aware visual-textual presentation layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6018–6026, 2023.

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

Zhaoyun Jiang, Jiaqi Guo, Shizhao Sun, Huayu Deng, Zhongkai Wu, Vuksan Mijovic, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. Layoutformer++: Conditional graphic layout generation via constraint serialization and decoding space restriction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18403–18412, 2023.

Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. LayoutVAE: Stochastic scene layout generation from a label set. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9895–9904, 2019.

Xiang Kong, Lu Jiang, Huiwen Chang, Han Zhang, Yuan Hao, Haifeng Gong, and Irfan Essa. Blt: Bidirectional layout transformer for controllable layout generation. In *Proceedings of the European Conference on Computer Vision*, pp. 474–490. Springer, 2022.

Hsin-Ying Lee, Lu Jiang, Irfan Essa, Phuong B Le, Haifeng Gong, Ming-Hsuan Yang, and Weilong Yang. Neural design network: Graphic layout generation with constraints. In *Proceedings of the European Conference on Computer Vision*, pp. 491–506. Springer, 2020.

Elad Levi, Eli Brosh, Mykola Mykhailych, and Meir Perez. DLT: Conditioned layout generation with joint discrete-continuous diffusion layout transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2106–2115, 2023.

Fengheng Li, An Liu, Wei Feng, Honghe Zhu, Yaoyu Li, Zheng Zhang, Jingjing Lv, Xin Zhu, Jun-Jun Shen, Zhangang Lin, and Jingping Shao. Relation-aware diffusion model for controllable poster layout generation. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 1249–1258, 2023.

Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. LayoutGAN: Generating graphic layouts with wireframe discriminators. In *Proceedings of the International Conference on Learning Representations*, 2019.

Jianan Li, Jimei Yang, Jianming Zhang, Chang Liu, Christina Wang, and Tingfa Xu. Attribute-conditioned layout GAN for automatic graphic design. *IEEE Transactions on Visualization and Computer Graphics*, 27(10):4039–4048, 2020.

Zhen Li, Duan Li, Yukai Guo, Xinyuan Guo, Bowen Li, Lanxi Xiao, Shenyu Qiao, Jiashu Chen, Zijian Wu, Hui Zhang, Xinhuan Shu, and Shixia Liu. ChartGalaxy: A dataset for infographic chart understanding and generation. In *Proceedings of the International Conference on Learning Representations*, pp. 52484–52546, 2026.

Jiawei Lin, Jiaqi Guo, Shizhao Sun, Zijiang Yang, Jian-Guang Lou, and Dongmei Zhang. LayoutPrompter: Awaken the design ability of large language models. In *Proceedings of the Conference on Neural Information Processing Systems*, pp. 43852–43879, 2023.

Jiawei Lin, Shizhao Sun, Danqing Huang, Ting Liu, Ji Li, and Jiang Bian. From elements to design: A layered approach for automatic graphic design composition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8128–8137, 2025.

Jiawei Lin, Shizhao Sun, Danqing Huang, Ting Liu, Ji Li, and Jiang Bian. ReLayout: Versatile and structure-preserving design layout editing via relation-aware design reconstruction. *arXiv preprint arXiv:2602.01046*, 2026.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. GDPO: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

OpenAI. Advancing the price-performance frontier with gpt-5.6, 2026a. URL https://openai.com/index/gpt-5-6/. Accessed: 2026-9-1.

OpenAI. Introducing chatgpt images 2.0, 2026b. URL https://openai.com/index/introducing-chatgpt-images-2-0/. Accessed: 2026-9-1.

Zhenyu Pan and Han Liu. MetaSpatial: Reinforcing 3d spatial reasoning in vlms for the metaverse. In *Proceedings of the International Conference on Learning Representations*, pp. 117974–117993, 2026.

Perceptual Hashing. Image deduplicator (imagededup), 2019. URL https://github.com/idealo/imagededup. Accessed: 2025-05-07.

PPTMON. Free google slide themes powerpoint templates, 2026. URL https://pptmon.com/. Accessed: 2026-9-1.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pp. 8748–8763, 2021.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

SlidesCarnival. Free powerpoint and google slides templates for your presentations, 2026. URL https://www.slidescarnival.com/. Accessed: 2026-9-1.

Zecheng Tang, Chenfei Wu, Juntao Li, and Nan Duan. LayoutNUWA: Revealing the hidden layout expertise of large language models. In *Proceedings of the International Conference on Learning Representations*, 2024.

Haoran Wang, Bo Zhao, Jinghui Wang, Hanzhang Wang, Huan Yang, Wei Ji, Hao Liu, and Xinyan Xiao. SEGA: A stepwise evolution paradigm for content-aware layout generation with design prior. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19321–19330. IEEE, 2025.

Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, Jan Kautz, Lei Zhang, and Zhiding Yu. LocateAnything: Fast and high-quality vision-language grounding with parallel box decoding. In *European Conference on Computer Vision*, pp. 336–357. Springer, 2026.

Jiazhe Wei, Ken Li, Tianyu Lao, Haofan Wang, Liang Wang, Caifeng Shan, and Chenyang Si. Postercopilot: Toward layout reasoning and controllable editing for professional graphic design. *arXiv preprint arXiv:2512.04082*, 2025.

Kota Yamaguchi. CanvasVAE: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5481–5489, 2021.

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran XU, Xinyu Zhu, Siheng Li, Yuxiang Zhang, Gongye Liu, Xiaomei Nie, Deng Cai, and Yujiu Yang. ChartMimic: Evaluating LMM's cross-modal reasoning capability via chart-to-code generation. In *Proceedings of the International Conference on Learning Representations*, pp. 26590–26646, 2025.

Junyi Zhang, Jiaqi Guo, Shizhao Sun, Jian-Guang Lou, and Dongmei Zhang. LayoutDiffusion: Improving graphic layout generation by discrete diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7226–7236, 2023.

Min Zhou, Chenchen Xu, Ye Ma, Tiezheng Ge, Yuning Jiang, and Weiwei Xu. Composition-aware graphic layout GAN for visual-textual presentation designs. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 4995–5001, 2022.

Jiangning Zhu, Bowen Li, Shenyu Qiao, Yima Gu, Zhao Zhang, Yuhui Yuan, and Shixia Liu. Detect anything in graphic design: Element-level rewards for autoregressive detection. *arXiv preprint arXiv:2609.07072*, 2026.

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

A Additional Qualitative Results of ReCanvas

Figs. 5 and 6 present additional qualitative results of ReCanvas. Across diverse source designs and target canvases, ReCanvas produces coherent adaptations while preserving the main content and structural characteristics of the source designs.

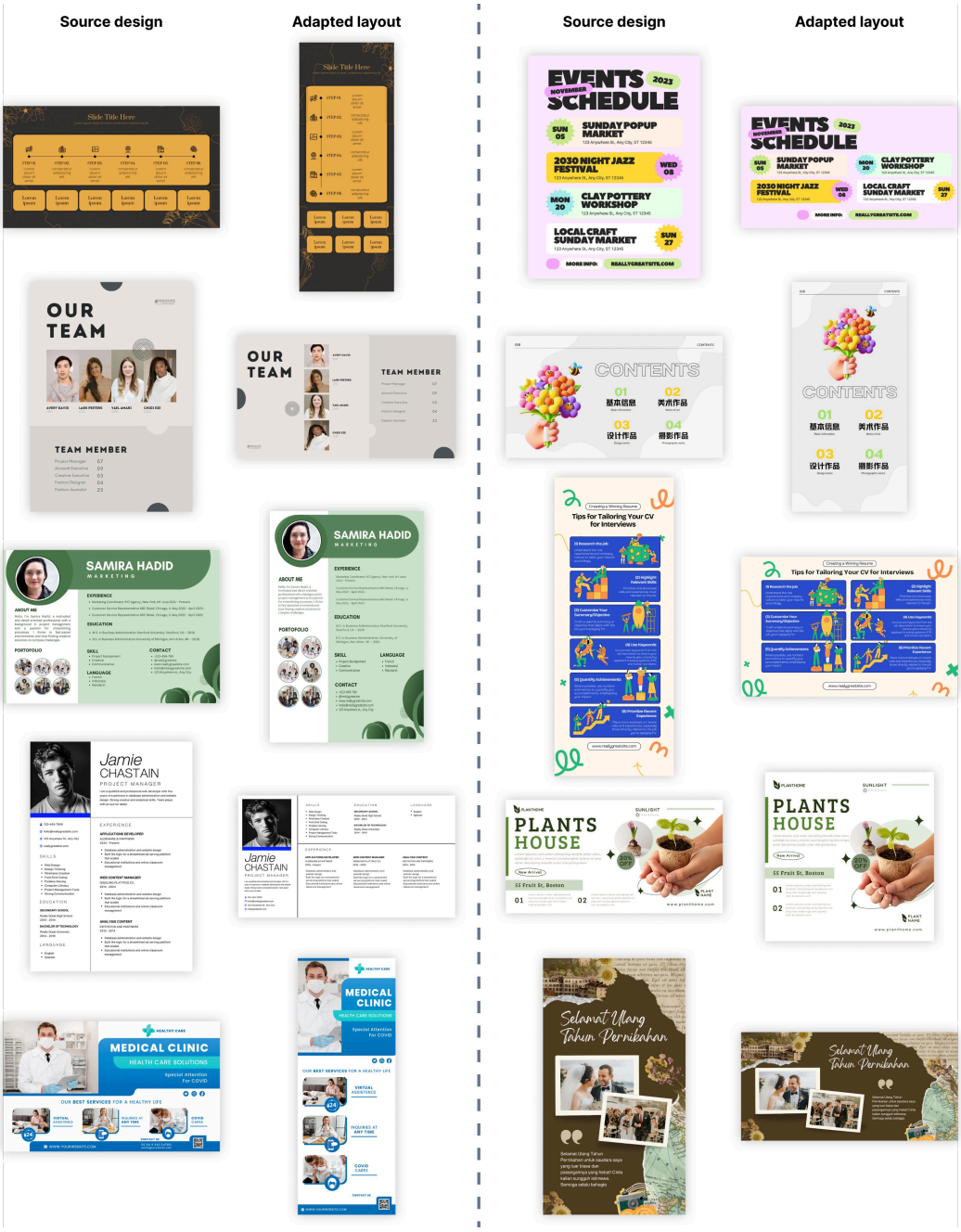

Figure 5: Additional qualitative results of ReCanvas (Part 1).



Figure 6: Additional qualitative results of ReCanvas (Part 2).

B Additional Details of ReCanvas-Data

B.1 Data Sources and Usage Terms

Crello (Yamaguchi, 2021) is a layered graphic design dataset derived from templates originally collected from Crello (now VistaCreate). It provides canvas information and element-level representations, including element geometry, rendered element images, text content, and typographic attributes. The Hugging Face release is distributed under CDLA-Permissive-2.0. Its dataset documentation addi-

Table 7: Data sources and their stated licenses. Additional source-specific terms may apply, as described below.

Name	URL	License
Crello	huggingface.co/datasets/cyberagent/crello	CDLA-Permissive-2.0
GenPoster-100K	huggingface.co/datasets/BruceW91/GenPoster-100K	CC BY-NC 4.0
PPTMON	pptmon.com	CC BY 4.0
SlidesCarnival	www.slidescarnival.com	CC BY 4.0
ChartGalaxy	huggingface.co/datasets/ChartGalaxy/ChartGalaxy	CC BY-NC 4.0

tionally states that the distributor does not own the copyrights of the original design templates, that users must comply with the applicable VistaCreate license agreements, and that the original template files are not redistributed. We use the released layered representation and retain its provenance when constructing RECANVAS-DATA.

GenPoster-100K (Wang et al., 2025) is a large-scale collection of poster designs with parseable layers and rich element-level annotations. Its Hugging Face release is labeled CC BY-NC 4.0. The source release further states that the data are provided for academic use and that commercial use requires permission from the copyright owner. We use GenPoster-100K only for non-commercial academic research and preserve its source attribution and license information.

PPTMON (PPTMON, 2026) provides editable PowerPoint and Google Slides presentation templates. Its templates are made available under CC BY 4.0 with attribution requirements. PPTMON additionally restricts redistribution of the original template files "as is" and notes that photographs included in the templates may be governed by separate licenses. Its terms also specify additional restrictions on derivative uses outside presentation purposes. We use the templates as source presentation designs for research and retain source-level provenance for each collected design.

SlidesCarnival (SlidesCarnival, 2026) provides editable presentation templates for PowerPoint and Google Slides under CC BY 4.0. The license permits sharing and adaptation with appropriate attribution, while the site's usage terms prohibit redistribution of the original template files "as is." SlidesCarnival also notes that photographs included in its templates may carry their own licenses. We therefore retain the provenance of the original template and treat embedded third-party assets according to the corresponding source terms.

ChartGalaxy (Li et al., 2026) contains both real and synthetically generated infographic charts. In this work, we use only synthetic infographic designs produced by the ChartGalaxy generation pipeline; we do not use the real charts collected from third-party websites as source designs for RECANVAS-DATA. The Hugging Face repository metadata lists CC BY-NC 4.0 for the dataset. We conservatively follow this non-commercial restriction for the data used in this work and retain the corresponding provenance information.

Because the above sources differ in their licensing and redistribution terms, we preserve source provenance and license metadata throughout our processing pipeline rather than applying a single license to all upstream content. Source-derived elements are redistributed only where permitted by the corresponding license and source-specific terms. For content whose redistribution rights are restricted or unclear, we do not redistribute the original assets and instead provide source references and processing instructions where permitted. Users of RECANVAS-DATA are responsible for complying with the licenses and usage terms of the corresponding upstream sources.

B.2 RECANVAS-DATA STATISTICS

Data distribution Table 8 reports the per-source statistics throughout the data construction pipeline. We collect 313,648 layered designs, of which 291,813 remain for layout-pair synthesis. From these designs, we generate 1,082,848 target variants before quality filtering and retain 896,505 (82.8%) after filtering, corresponding to an average of 3.07 accepted variants per source design. Together with the source layouts, these variants form 4,222,410 directed source–target adaptation pairs.

Fig. 7 further summarizes the distributions of design complexity and canvas change. We measure design complexity by the number of elements in each of the 291,813 source designs. Most designs

Table 8: Per-source statistics of the data processing pipeline.

Source	Collected Designs	Processed Designs	Generated Variants	Accepted Variants	Directed Pairs
ChartGalaxy	127,959	119,769	359,235	290,129	1,074,870
GenPoster-100K	104,784	101,052	403,690	310,750	1,515,356
PPTMON	37,670	33,827	169,128	160,339	939,780
Crello	22,170	16,975	50,365	43,770	166,176
SlidesCarnival	21,065	20,190	100,430	91,517	526,228
Total	**313,648**	**291,813**	**1,082,848**	**896,505**	**4,222,410**

contain 6–20 elements (83.9%), while 11.7% contain more than 20 elements and 0.7% contain more than 40. For a directed pair with source and target aspect ratios r_s and r_t, respectively, we define the aspect-ratio change factor as $\max(r_s/r_t, r_t/r_s)$. The resulting pairs cover a broad range of canvas changes: 73.5% have an aspect-ratio change greater than $1.5\times$, 46.9% greater than $2\times$, and 6.8% greater than $4\times$.

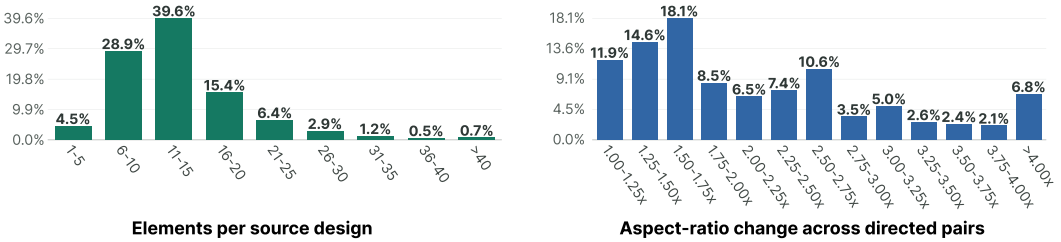

Figure 7: Distributions of elements per source design and aspect-ratio change across directed pairs in RECANVAS-DATA.

Target aspect ratios We define design-type-specific candidate sets of target aspect ratios to cover diverse canvas shapes encountered in real-world graphic design. As listed in Table 9, posters and infographics share a set of 31 aspect ratios spanning portrait, square, landscape, and more extreme canvas shapes, while presentation slides use 9 commonly adopted presentation aspect ratios. For each source design, we select multiple target canvases from the corresponding candidate set, favoring ratios that differ sufficiently from the source and from one another, with a minimum aspect-ratio change factor of 1.3 where applicable. The assignment is balanced across candidate ratios to avoid concentrating generated variants on a small number of canvas shapes. We additionally generate more target variants for presentation slides to improve the balance across design types. This sampling procedure produces the 1,082,848 candidate target variants reported in Table 8.

Table 9: Candidate target aspect ratios for each design type.

Design Type	# Ratios	Candidate Aspect Ratios
Poster / Infographic	31	1:3, 8:21, 2:5, 52:119, 1:2, 21:40, 9:16, 7:12, 3:5, 5:8, 2:3, 22:31, 3:4, 4:5, 5:6, 1:1, 6:5, 5:4, 4:3, 31:22, 3:2, 8:5, 5:3, 12:7, 16:9, 40:21, 2:1, 119:52, 5:2, 21:8, 3:1
Slide	9	9:16, 2:3, 22:31, 4:3, 31:22, 3:2, 16:10, 16:9, 21:9

B.3 EXAMPLES OF RECANVAS-DATA

Fig. 8 presents examples of source–target layout pairs in RECANVAS-DATA. The paired layouts capture diverse adaptations across canvas shapes, including layout-level changes in element placement and scale, as well as element-level transformations such as rotation, text reflow, image cropping, rotation, and flipping. These examples illustrate how RECANVAS-DATA preserves the content and

overall design structure while allowing the transformations necessary for adaptation to different canvases. Overflowed elements outside the visible canvas boundary are shown under a shaded overlay.



Figure 8: Examples of source–target layout pairs in RECANVAS-DATA.

C ADDITIONAL DETAILS OF RECANVAS-BENCH

C.1 RECANVAS-BENCH STATISTICS

Benchmark composition RECANVAS-BENCH contains 500 held-out source designs, with 100 designs sampled from each of the five data sources. Each source design is paired with three target

canvases, yielding 1,500 adaptation tasks in total. As shown in Fig. 9, the benchmark covers a broad range of design complexity, with element counts ranging from 4 to 50. The element-count distributions vary across data sources, reflecting their different design characteristics rather than enforcing an identical distribution for each source.

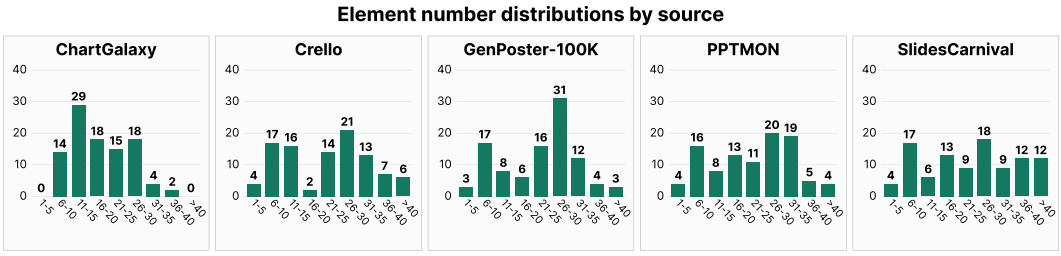

Figure 9: Element-count distributions of the 500 source designs in RECANVAS-BENCH, shown separately for each data source.

Canvas-shape transitions Fig. 10 summarizes the source–target canvas-shape transitions in the benchmark. We categorize canvases into five shape classes: extreme portrait (EP), portrait (P), square (S), landscape (L), and extreme landscape (EL). Aspect ratios more extreme than 9:16 in portrait orientation or 16:9 in landscape orientation are categorized as EP and EL, respectively. Each cell in Fig. 10 reports the number of adaptation tasks for a source–target shape transition and its percentage within the corresponding source-shape row.

The source-shape distributions reflect the native formats of the underlying data sources. For example, all PPTMON source slides use a 16:9 canvas and therefore fall into the landscape category, resulting in a single non-empty source row. In contrast, the sampled target canvases span a wider range of shapes. Square targets occur less frequently because the candidate set contains only one square aspect ratio, whereas the portrait and landscape categories each cover multiple candidate ratios.

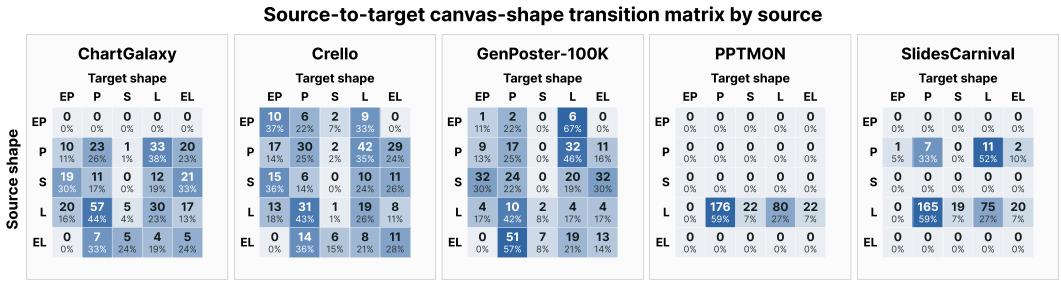

Figure 10: Source–target canvas-shape transitions in RECANVAS-BENCH, shown separately for each data source. Each cell reports the number of tasks and the percentage within the corresponding source-shape row. EP, P, S, L, and EL denote extreme portrait, portrait, square, landscape, and extreme landscape, respectively.

Aspect-ratio change levels Fig. 11 further breaks down canvas-shape transitions by the three aspect-ratio change levels used in RECANVAS-BENCH. Each level contains 500 tasks, corresponding to change factors of 1–1.5×, 1.5–2.5×, and $> 2.5\times$ for small, medium, and large changes, respectively. Small changes predominantly preserve the source orientation and often remain within the same or a neighboring shape category. Medium changes involve more frequent transitions across shape categories, while large changes typically require substantial changes in canvas orientation, such as from portrait to landscape or vice versa. This distribution ensures that RECANVAS-BENCH evaluates adaptation across progressively more challenging canvas changes rather than only variations within similar canvas shapes.

C.2 EVALUATION METRICS OF RECANVAS-BENCH

We evaluate each predicted layout with respect to its source design rather than a paired target layout, since multiple adaptations can be valid for the same target canvas. Our evaluation combines

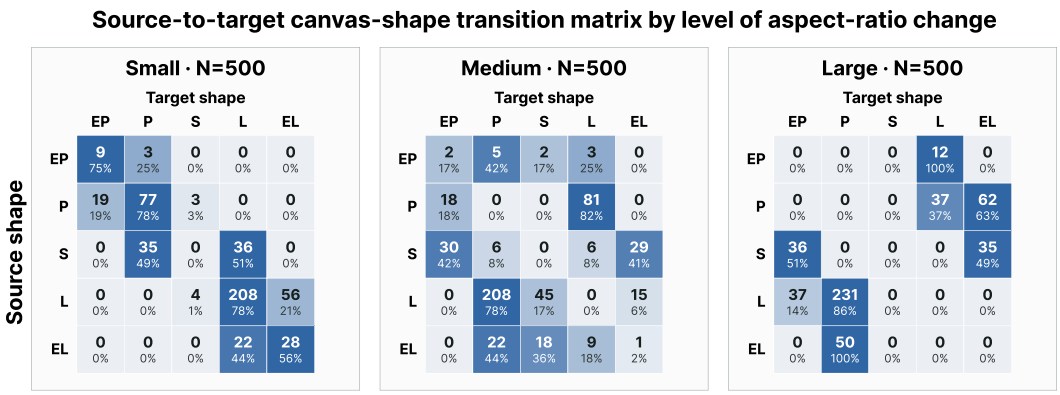

Figure 11: Source–target canvas-shape transitions at different levels of aspect-ratio change in RECANVAS-BENCH. Each level contains 500 tasks. Cell percentages are computed within each source-shape row.

programmatic metrics computed from element geometry and rendered masks with a perceptual evaluation of the rendered result. Unless otherwise specified, all scores are first computed per task and then averaged over the 1,500 tasks in RECANVAS-BENCH. Outputs that cannot be parsed or do not contain exactly one prediction for every input element are treated as invalid and receive a score of zero on all metrics. For geometric computation, element transformations are applied in the order of horizontal flip, rotation, crop, placement, and z-order compositing.

Layout readability We evaluate readability through *Element Visibility* and *Text Readability*. For each element e_i, Element Visibility requires both (1) at least 10% of its predicted bounding-box area to lie inside the target canvas and (2) at least 10% of its rendered support to remain visible after accounting for occlusion by elements above it. Let v_i^{canvas} and v_i^{occ} denote these two fractions. We define

$$V_i = \mathbb{I}[v_i^{\text{canvas}} \geq 0.1]\mathbb{I}[v_i^{\text{occ}} \geq 0.1], \qquad R_{\text{elem}} = \frac{1}{N_e}\sum_i V_i. \tag{4}$$

Text Readability evaluates whether each text element remains visible, legible, and properly rendered after adaptation. To avoid penalizing intentional typographic effects already present in the source design, each readability check is source-conditioned: if the source text already violates the corresponding criterion, that check is waived for the target. Otherwise, we check text visibility, canvas overflow, minimum rendered size, and contrast against the composited background. The Text Readability score is the fraction of text elements satisfying all applicable checks; designs without text are considered to pass this metric.

We additionally report *All*, a strict case-level readability score. A task passes only when every element satisfies Element Visibility and every text element satisfies Text Readability. Thus, unlike the individual scores, a single unreadable text element or insufficiently visible element causes the entire task to fail the aggregated readability check.

Source similarity We further evaluate how well the adapted layout preserves source-derived layout characteristics using five complementary metrics: alignment, relative size, occlusion, containment, and space utilization. These metrics compare the predicted target layout with the source layout without requiring their coordinates to remain unchanged.

Alignment. We normalize bounding boxes by the corresponding canvas dimensions and consider six anchors for each element: left, horizontal center, right, top, vertical center, and bottom. An element participates in an alignment if at least one corresponding anchor is within 1% of the canvas dimension of an anchor from another element. Let $\mathcal{A}_s$ and $\mathcal{A}_t$ denote the sets of aligned elements in the source and target layouts. We compute $R_{\text{align}} = \text{F1}(\mathcal{A}_s, \mathcal{A}_t)$. We intentionally do not require the same alignment partner or axis to be preserved, allowing the layout to establish new alignments when adapting to a different canvas.

Relative size. For every element pair, we compare their canvas-normalized, clipped bounding-box areas and classify their relative-size relation as *larger*, *smaller*, or *equal*, where areas within 10% are considered equal. For each element, we compute the macro-F1 of these relations between the source and target layouts and average the resulting scores across elements.

Occlusion. We use rendered element masks and z-order to construct directed occlusion relationships. For each element e_i, we record the elements it occludes and those by which it is occluded. The source and target occlusion neighborhoods are compared using F1, and the resulting scores are averaged over all elements. This metric therefore captures both whether elements overlap and whether their front–back ordering is preserved.

Containment. We evaluate containment for elements identified as semantic containers or badges. An element is considered contained when at least 95% of its rendered support lies within an eligible outer element, with the outer element appearing below it in z-order and having at least $1.05\times$ its support area. Containment relationships in the source and target layouts are compared using F1. We average the containment F1 over all eligible source containers in the benchmark.

Space utilization. We measure the fraction of the canvas occupied by the union of foreground element bounding boxes, excluding backgrounds and containers that could trivially cover large portions of the canvas. Let u_s and u_t denote the resulting utilization ratios of the source and target layouts. We define

$$R_{\text{space}} = \min\left(\frac{u_s}{u_t}, \frac{u_t}{u_s}\right), \tag{5}$$

with a score of 1 when both ratios are zero and 0 when exactly one is zero. This metric measures preservation of the overall amount of occupied space rather than requiring identical spatial arrangements.

The aggregated similarity score reported in the main experiments is the unweighted average of the five metrics:

$$R_{\text{sim}} = \frac{1}{5}\left(R_{\text{align}} + R_{\text{size}} + R_{\text{occ}} + R_{\text{cont}} + R_{\text{space}}\right). \tag{6}$$

VLM-based evaluation The programmatic metrics capture explicit geometric and readability properties but cannot fully characterize visual hierarchy, balance, coherent reflow, and overall composition quality. We therefore complement them with a VLM-based perceptual evaluation using Gemini-3.6-Flash (Google, 2026). For each task, the evaluator receives only the rendered source design and the rendered model output; it does not receive the paired target layout, element metadata, or programmatic metric values. Each output is evaluated independently on four dimensions: *Element Fidelity* (0–20), which assesses content completeness and the visual integrity of non-text elements; *Text Readability* (0–20), which assesses textual accuracy, typography, wrapping, contrast, and legibility; *Structural Preservation* (0–30), which assesses hierarchy, reading order, grouping, alignment, relative size, containment, overlap, and spacing; and *Composition Quality* (0–30), which assesses canvas utilization, visual balance and rhythm, and overall finish and coherence. The four scores sum to an overall VLM score between 0 and 100. We provide the prompt in Appendix G.5.

D Model Implementation Details

D.1 RGBA Visual Encoding

ReCanvas is initialized from Qwen3.5-9B. The original vision encoder uses a spatiotemporal convolution as its patch embedding, with weights

$$W_{\text{RGB}} \in \mathbb{R}^{d\times 3\times 2\times 16\times 16}, \tag{7}$$

where $d = 1152$ for Qwen3.5-9B. To preserve the transparency information of layered graphic elements, we extend the input from RGB to RGBA:

$$W_{\text{RGBA}} \in \mathbb{R}^{d\times 4\times 2\times 16\times 16}. \tag{8}$$

This modification adds approximately 0.59M parameters while leaving the number and dimensionality of visual tokens and all subsequent vision and language layers unchanged. We initialize the new

RGBA patch embedding by copying the pretrained RGB weights and setting the additional alpha-channel weights to zero:

$$W_{\text{RGBA}} = [W_R;, W_G;, W_B;, \mathbf{0}_A]\,. \tag{9}$$

Consequently, the modified model initially preserves the behavior of the pretrained RGB model, while learning to use transparency information during fine-tuning.

We further modify the image preprocessing pipeline to preserve alpha values rather than converting all inputs to RGB. RGBA inputs use the original RGB normalization for the first three channels and normalize the alpha channel with mean 0.5 and standard deviation 0.5. Images without transparency are assigned a fully opaque alpha channel. Thus, rendered source designs can be processed as ordinary opaque images, while individual transparent image layers retain their original alpha masks.

D.2 Supervised Training Details

The supervised training of ReCanvas includes two stages, as described in Sec. 4.3.

In LG-SFT, we perform full-parameter layout-generation training for 100K optimizer steps with a global batch size of 128. We use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, a weight decay of 0.1, and a learning rate of 2×10^{-5}. The learning rate is linearly increased during the first 5K steps and then kept constant.

In LA-SFT, we initialize the model from the 100K-step LG-SFT checkpoint and perform full-parameter supervised fine-tuning on the source-conditioned layout-adaptation task for an additional 20K steps. The optimizer and learning-rate scheduler are reinitialized. We use a global batch size of 128, a peak learning rate of 2×10^{-5}, and a cosine schedule that decays the learning rate to zero after 1K warmup steps.

We set the maximum multimodal sequence length to 8,192 tokens and discard samples exceeding this limit rather than truncating their target outputs. The maximum image size is 2,097,152 pixels, corresponding to at most 2,048 merged visual tokens per image under the Qwen3.5 vision tokenizer. In LA-SFT, the source-design image is resized to a maximum long edge of 512 pixels, while individual RGBA layer images are resized to a maximum long edge of 128 pixels with aspect-ratio preservation and padding to multiples of 32. Gradients are clipped to a global norm of 1.0. Both stages are trained on 16 NVIDIA H200 GPUs with a per-GPU micro-batch size of 2 and gradient-accumulation step of 4. All controlled LG-SFT and LA-SFT experiments on ReCanvas-Data follow the same training parameters.

D.3 Reinforcement Learning Details

Training setup We initialize reinforcement learning from the adaptation-SFT checkpoint. For each input, we sample $G = 8$ completions and optimize the policy using the GRPO-style objective with GDPO-normalized multi-reward advantages described in Sec. 4.3. We train for 1,000 steps with 32 prompts per global batch and 8 rollouts per prompt, yielding 256 sampled completions per batch. We use a constant learning rate of 10^{-6} and a maximum completion length of 8,192 tokens. Rollouts are sampled with temperature 1.0 and top-p 0.95. We use a clipping threshold of $\epsilon_c = 0.2$ and a KL coefficient of $\beta = 0.01$.

Reward design As described in Sec. 4.3, we use four verifiable rewards and one VLM-based perceptual reward. The element, text, and relation rewards, R_{ele}, R_{text}, and R_{rel}, are derived from the same criteria used by the programmatic evaluation metrics in Appendix C.2. They respectively assess element visibility and integrity, text readability, and preservation of source-derived layout relationships. Below, we provide additional details on the scene-tree and perceptual-quality rewards.

Scene-tree reward. R_{tree} evaluates the intermediate scene tree predicted by LS-CoT against the source scene tree. We compare the two trees in three aspects: (1) semantic consistency of the element categories, (2) hierarchical grouping, measured by matching the element sets represented by internal groups, and (3) element ordering, measured by pairwise order agreement. The resulting similarity provides a verifiable reward directly on the intermediate structural reasoning, independently of the final layout coordinates.

Perceptual-quality reward. R_{qual} complements the verifiable rewards with a holistic assessment of the rendered adaptation. For each input, we render all valid rollouts and present them together with

the source design to Gemini-3.6-Flash (Google, 2026). The candidates are anonymized and randomly ordered. Rather than matching a target reference, the judge evaluates whether each candidate forms a faithful and well-composed adaptation of the source. It first considers major validity and content-preservation failures, including missing or duplicated elements, destructive occlusion or cropping, corrupted text, and loss of important information. Among otherwise valid candidates, it further considers visual hierarchy, alignment, spacing, balance, and utilization of the target canvas. The candidates are scored jointly on a shared 0–100 scale so that outputs of similar quality are further distinguished by these compositional criteria. The resulting score is used as R_{qual}. We provide the system prompt in Appendix G.4.

E Scene-Tree Annotation Details

Annotation procedure We use Gemini-3.6-Flash (Google, 2026) to annotate scene trees for the source designs. Given the rendered design together with its layered elements and layout information, the model is prompted to assign a semantic category to every visible element and organize all elements into an ordered hierarchical tree. The annotation prompt specifies structural constraints including complete element coverage, unique leaf assignment, non-trivial grouping, and ordered siblings, while emphasizing that semantic categories should not determine the grouping structure. The complete annotation prompt is provided in Appendix G.3.

Scene-tree representation Each visible element appears exactly once as a leaf node, while internal nodes represent groups of elements that form meaningful visual or semantic units. Internal group nodes are intentionally left unlabeled and contain at least two descendant elements. Sibling nodes follow the intended reading or design order of the source composition. Each leaf is additionally assigned one semantic category from the taxonomy in Table 10. The semantic category describes the functional role of an individual element and is independent of the grouping hierarchy.

Grouping principles The annotation prioritizes information organization over geometric proximity or semantic category. We first identify local units that should remain together, such as heading blocks, icon–text pairs, media–caption pairs, badges, and container–content units. Related local units are then recursively grouped into larger information blocks, repeated structures, columns, sections, or page regions. Decorative elements are attached to the content unit they visually support when such an association is clear. Importantly, elements with different semantic categories may belong to the same group, while elements sharing the same category are not necessarily grouped. This prevents the semantic taxonomy from determining the tree structure.

Semantic taxonomy The taxonomy describes the coarse functional role of each element and is organized into four high-level categories: text, media, background, and graphic. Within each category, finer-grained labels distinguish common roles in graphic designs, as summarized in Table 10. These labels provide auxiliary semantic information for scene-tree prediction; the hierarchy itself is determined by the grouping principles above.

F Agreement Analysis of Human and VLM Judgments

In the human evaluation, participants view the source design alongside two candidate adaptations and rate their preference on a three-level scale: the first better, the second better, or the same. Model identities are hidden, and both question order and left–right placement are randomized.

All three raters assign the same directional label to 439 of the 500 pairs (87.8%). Pairwise human–human agreement rates are 93.2%, 91.0%, and 90.0%, with corresponding Cohen's κ values of 0.886, 0.846, and 0.830, respectively. The overall Fleiss' κ is 0.854, indicating high inter-rater agreement. We derive the VLM preference from the relative scores of the two outputs, treating exactly equal scores as a tie.

On the 439 pairs with unanimous human judgments, the VLM agrees with the human consensus on 364 pairs (82.9%), corresponding to Cohen's $\kappa = 0.704$. For the remaining 61 pairs, where the human raters disagree, pairwise human–human agreement drops to 44.3%, 26.2%, and 18.0% (29.5% on average). On this subset, VLM–human agreement with the three individual raters is 41.0%, 32.8%, and 47.5%, respectively, yielding 40.4% agreement across the 183 individual judgments. These

Table 10: Semantic taxonomy used for scene-tree annotation. Each leaf element is assigned exactly one category.

Type	Code	Category	Description
Text	tt	Title	Primary headline or section title.
	ts	Subtitle	Secondary title or supporting headline.
	tb	Body Text	Paragraphs, descriptions, instructions, and other body content.
	tc	Caption	Text associated with nearby media, charts, or other content.
	tm	Metadata	Dates, locations, URLs, social handles, and other auxiliary information.
	tl	Label	Labels for categories, legends, lists, menus, or similar items.
	tn	Hero Number	Visually prominent numbers such as prices, countdowns, or step numbers.
	tw	Badge Text	Text displayed within or together with a badge or tag.
Media	mh	Hero Media	Primary photo, illustration, product, or other dominant visual.
	ms	Secondary Media	Supporting photos, illustrations, maps, diagrams, or screenshots.
	ml	Logo	Logo marks, wordmarks, or brand symbols.
	mx	Decorative Media	Decorative artwork, ornaments, textures, or visual accents.
Background	mb	Background Media	Full-page or sectional background imagery or texture.
Graphic	gv	Data Visualization	Charts, tables, and other graphics encoding structured data.
	gb	Badge	Non-text badges, pills, tags, or marker shapes.
	gc	Container	Panels, cards, frames, buttons, or other shapes containing content.
	ge	Emphasizer	Callouts or selection marks emphasizing a specific element.
	gl	Line	Arrows, connectors, underlines, rules, or targeted accent lines.
	gi	Icon	Semantic non-logo icons such as location, calendar, or contact icons.

results show that the VLM evaluator is broadly consistent with human preferences when human judgments are themselves consistent, while most disagreement is concentrated in cases with low human consensus.

G System Prompts

G.1 Image Editing Prompt for Layout Adaptation

```
Relayout the existing design defined by the four neon-green L-shaped
corner markers to a {target_ratio} canvas at {target_size}.

INPUT SEMANTICS
- Only the four pure neon-green (#00FF00) L-shaped corner markers are
metadata, not artwork. Together they identify the authoritative
original canvas. Do not include these four markers in the output.
- All other green, mint, teal, cyan, and gradient content is original
artwork and must be preserved. Do not remove or recolor it because it
is green.
- Content inside the four corner markers is the authoritative original
 design.
- Content outside that rectangle is bounded overflow reference only.
Use it only to recover existing portions of elements already visible
in the original design.
- Do not treat the full source map as the original composition.

ALLOWED
- Reposition, crop, scale, reasonably stretch, rotate, or flip
existing visual content.
- Background pixels may be cropped, scaled, or reasonably stretched to
 cover the target canvas.
```

```
PRESERVE
- Preserve the original information organization: semantic groups,
relationships within and between groups, reading order, emphasis, and
visual hierarchy.
- Keep content that belongs together grouped together. Preserve the
original headline-to-supporting-content order and the relative
prominence of primary, secondary, and supporting information, even
when their spatial arrangement changes.
- Preserve the recognizable identity, appearance, palette, effects,
and internal visual content of the original design.
- Preserve only text visibly present inside the marked original canvas
. Keep its wording exactly; line breaks may change only when needed
for the new layout.
- Shapes or panels that are blank in the original design must remain
blank.

FORBIDDEN
- No outpainting, inpainting, content-aware fill, reconstruction,
redrawing, restyling, or generative completion.
- Do not add, duplicate, replace, or hallucinate objects or text.
- Do not reveal hidden, invisible, masked, or metadata-only text.

Produce an intentional {target_ratio} relayout, not an inset copy of
the old canvas.
Return only the final image.
```

G.2 Quality Filtering Prompt

```
You are evaluating whether a reconstructed relayout is usable training
 data.

You will receive exactly two images in this order:
1. SOURCE: the original design rendered from the available source
layers.
2. RELAYOUT: the final reconstruction after those layers were
rearranged for a different canvas.

Judge the RELAYOUT both as a standalone design and as a faithful
relayout of SOURCE. The canvas size and aspect ratio may change
substantially. A good relayout is not expected to be pixel-identical
or to keep the original absolute positions.

Evaluate these four checks:

LAYOUT
- The relayout should remain understandable and visually usable.
- Important text and media should have sensible scale, spacing,
hierarchy, alignment, and reading order.
- Reject severe crowding, fragmentation, huge unintended empty regions
, implausible scaling, or important content placed outside the canvas.
- Mild imbalance, imperfect spacing/alignment, changed line wrapping,
or small typography differences are acceptable.

COMPLETENESS
- Compare visible semantic units rather than raw pixels: titles,
paragraphs, labels, numbers, logos, icons, photos, illustrations,
charts, containers, and meaningful decoration.
- Important source units must remain visible exactly once in the
relayout. Severe omissions or duplications are failures.
```

```
- A small nonessential ornament may be missing, simplified, or less
visible and still be a minor issue.
- Judge only what can be verified visually. Do not invent a missing
layer when an item may simply be transformed, cropped, or merged
visually.

INFORMATION CONSISTENCY
- Preserve the same wording, numbers, labels, branding, media identity
, and overall message.
- Reflow, line wrapping, resizing, rotation, and repositioning are
allowed.
- Minor rasterization, antialiasing, font-rendering, and color
differences are acceptable.
- Changed or unreadable key text/numbers, substituted media, broken
label-to-item association, or a materially changed message are serious
.

OCCLUSION AND CLIPPING
- Intentional overlap in SOURCE may remain or be adapted.
- The relayout must not introduce severe accidental occlusion, z-order
 inversion, text collision, or clipping that hides important
information.
- Be tolerant of layers that were already clipped at a canvas edge in
SOURCE. After an aspect-ratio change, an edge decoration or background
 layer may be cropped in a visually odd way; treat that as acceptable
or at most a minor issue unless it hides important text, a face/
product subject, a logo, chart data, or another key semantic unit.
- A small local overlap or slightly clipped nonessential decoration is
 a minor issue, not a failed reconstruction.

Do not fail a sample merely because the composition changed, the
target is not aesthetically optimal, or a few small defects exist.
Compare against SOURCE: do not blame RELAYOUT for a defect already
present in SOURCE unless it becomes materially worse.

Assign exactly one grade:
- grade 1, failed: reconstruction is unusable. It has at least one
major defect such as widespread layout collapse, important information
 missing/duplicated/changed, key content unreadable, or severe
accidental occlusion/clipping.
- grade 2, acceptable_with_minor_issues: reconstruction remains valid
and useful, but has one or more limited defects that do not materially
 prevent understanding or preserve-content use.
- grade 3, clean: no meaningful defect is visible. It may still differ
 substantially from SOURCE because it is a relayout.

‘acceptable‘ must be false only for grade 1 and true for grades 2 and
3.

For each check, use exactly one of: pass, minor_issue, major_issue,
uncertain.
- Grade 1 requires at least one major_issue.
- Grade 3 requires every check to be pass and issues to be empty.
- Use uncertain only when the images genuinely do not provide enough
visible evidence; uncertainty alone should normally produce grade 2,
not grade 1.

Issue codes must come from this list:
broken_layout, off_canvas_or_clipping, bad_scale,
bad_spacing_or_alignment, unreadable_text, text_collision,
missing_element, duplicate_element, changed_text_or_number,
changed_media, broken_grouping_or_reading_order, wrong_occlusion,
other.
```

```
Output exactly one JSON object with these keys and no additional keys:
 grade, acceptable, checks, issues, summary, confidence.
- checks must contain exactly: layout, completeness, information,
occlusion.
- Each issues item must contain exactly: code, severity, region,
detail.
- Use at most five issues. Severity must be minor or major. Confidence
 must be high, medium, or low.
- Keep region, detail, and summary concise and evidence-based.
- Do not output markdown, commentary, or chain-of-thought.
```

G.3 Scene Tree Annotation Prompt

```
You annotate graphic designs with a concise scene tree and exactly one
 semantic slot for every visible layer.
Output exactly one JSON object. No markdown, prose, comments, or code
fence.

Primary task:
1. Assign exactly one slot code to every visible layer.
2. Build one nested top-down grouping tree that captures meaningful
information organization and visual keep-together units.
3. Preserve local semantic units, their attached decoration, larger
information blocks, and page reading order during resize or relayout.

Grouping correctness has priority over perfect slot classification.
A slot classifies one layer. It must never be used as a reason to
group distant layers.
Grouping only decides which layers belong to one unit; it does not
classify a relation as relative or absolute and does not output
geometry relations.

Output schema:
{
  "items": [item, ...]
}

An item is exactly one of:
1. A single layer item:
   {"idx":4,"slot":"tt"}
2. A group item:
   {"items":[item, ...]}

Schema rules:
- Output no keys except "items", "idx", and "slot".
- A layer item has exactly one integer "idx" and exactly one "slot".
- A group item has exactly one key, "items". It has no id, slot, type,
 tag, label, text, layers, or metadata.
- Never put multiple layer ids in one item. Never output "layers" or
an array-valued "idx".
- Every visible layer id must occur exactly once in the complete tree.
- Use only ids listed in layers_norm1000.
- A group must contain at least two descendant layers. Never output an
 empty or one-layer group.
- Never wrap one layer in a group merely to avoid an ungrouped leaf. A
 truly standalone layer stays directly in its parent.
- If a group contains only one child group and nothing else, inline
that child group into the parent.
- Preserve reading/design order within every "items" list.
```

```
Valid:
{"items":[{"idx":0,"slot":"mb"},{"items":[{"idx":3,"slot":"gl"},{"idx
":4,"slot":"tt"},{"idx":5,"slot":"ts"},{"idx":6,"slot":"tb"},{"idx
":7,"slot":"mx"}]}]}

Invalid:
{"items":[{"slot":"tb","layers":[28,31,32]}]}
{"items":[{"idx":[28,31],"slot":"tb"}]}
{"items":[{"idx":28,"slot":"tb","text":"..."}]}
{"items":[{"items":[{"idx":4,"slot":"tt"}]}]}
{"items":[{"items":[{"idx":14,"slot":"mx"},{"idx":15,"slot":"mx"}],"
slot":"mx"}]}

Valid nested region pattern:
{"items":[
  {"idx":0,"slot":"mb"},
  {"items":[
    {"idx":1,"slot":"tm"},
    {"items":[{"idx":2,"slot":"tt"},{"idx":3,"slot":"tb"},{"idx":4,"
slot":"mx"}]},
    {"items":[{"idx":5,"slot":"mx"},{"idx":6,"slot":"tm"}]}
  ]},
  {"items":[{"idx":7,"slot":"mx"},{"idx":8,"slot":"mh"},{"idx":9,"slot
":"mx"}]}
]}
The middle object is one larger information region that preserves a
heading/decor unit and a footer unit as nested child groups. Do not
flatten those local units into six direct leaves.

Slot codes:
Use exactly the following codes.

TEXT:
- tt = Title: primary headline or section-title text.
- ts = Subtitle: secondary supporting title line under or near a title
.
- tb = Body Text: paragraphs, instructions, descriptions, and ordinary
 readable text.
- tc = Caption: text tied to a nearby image, chart, item, or media
element. Use position as evidence: text close to its media is usually
tc; a standalone paragraph is tb.
- tm = Metadata: date, time, location, URL, social handle, and other
small auxiliary facts.
- tl = Label Group Text: labels in menus, legends, categories, chips,
contact lines, feature lists, or list items.
- tn = Hero Number: a large or visually dominant number, including a
countdown, price, or step number used in the main visual hierarchy.
Price text is tn, not tm.
- tw = Text with Badge: text inside, on, or overlapping a badge, pill,
 tag, label shape, or its own backing color/shape. Plain label text
without such a backing is tl. A dominant price/countdown/step number
remains tn even when it has a backing shape.

MEDIA:
- mh = Hero Media: the main photo, product, illustration, or large
visual subject.
- ms = Secondary Media: supporting photos, thumbnails, smaller
illustrations, maps, ordinary diagrams, or screenshots.
- ml = Logo: a logo mark, wordmark, brand symbol, or sponsor mark.
- mx = Deco Media: a decorative ornament, texture, prop, sticker
artwork, abstract illustration, flourish, sparkle, confetti, or
standalone decorative shape used mainly for mood, framing, or visual
```

```
balance. Use mx when it does not point to, contain, label, or directly
 emphasize one specific content unit. An accent shape with no specific
 target is mx.

BACKGROUND:
- mb = Background Media: a full-page or section background photo,
texture, or illustration behind foreground information.

GRAPHIC:
- gv = Chart / Table / Data Visualization: a complete chart, table, or
 other data-encoding graphic layer, including plots, bars, lines, pie/
donut slices, axes, table grids/cells, and other marks that present
structured data. A generic grid layout that does not present data is
not a table. Ordinary illustrations, maps, diagrams, and screenshots
remain MEDIA.
- gb = Badge: a non-text badge, pill, tag, or marker shape. A
decorative sticker illustration is mx, not gb. If the badge contains
text, group the shape with that text; the text is normally tw.
- gc = Container: a panel, card, frame, mask, section background,
button, block, or other shape that visibly holds content. Content must
 be contained by, placed on, or substantially overlap the shape. A
nearby accent bar, stripe, rule, or decorative block that contains no
content is not gc. A divider whose purpose is to separate sections is
gc.
- ge = Emphasiser: a callout ring/circle, selection box, or selection
mark attached to one specific nearby target. Do not use ge for general
 decoration or line-shaped marks.
- gl = Line: a thin or thick elongated emphasis or connector such as
an arrow, pointer, leader line, underline, highlight line, accent bar,
 stripe, rule, or stroke. It must directly mark, frame, or emphasize
one specific content unit and be grouped with that target. A section-
separating divider is gc, not gl; an untargeted ambient accent is mx.
- gi = Icon: a semantic non-logo icon such as phone, mail, location,
calendar, social, bullet, or category icon.

Required grouping workflow:
1. Read the whole design and identify its major semantic regions and
intended reading flow.
2. Within each region, create the smallest meaningful local units:
heading blocks, text blocks, icon/marker lines, badge text, media-
caption units, chart units, logo units, and container-content units.
3. Merge local units that jointly communicate one information block,
such as heading plus supporting body, section title plus its list, or
product/media plus its caption and badge.
4. Attach nearby decoration and support graphics to the specific unit
or region they visually serve.
5. Wrap related local units into larger cards, repeated structures,
columns, sections, and page regions.
6. Audit every ungrouped layer leaf, especially direct children of
root. Keep it ungrouped only when no nearby semantic or visual unit
genuinely owns it.
7. Assign one slot to each layer without changing the grouping to fit
the slots, verify coverage, and output only the final tree.

Semantic grouping rules:
- Information organization comes first. Text, media, markers, and
decoration that jointly communicate one message should form one
information group even when they use different slots.
- A title, subtitle, and immediately supporting body/caption/metadata
form one heading or information block when they share alignment, width
, spacing, or meaning.
- A section heading belongs in the same larger section group as the
list, cards, chart, media, or body content it introduces.
```

```
- A marker, bullet, icon, badge, price/step number, or line normally
identifies one information unit. Pair it with its own nearby text/
content before building a larger list or section.
- Logo mark plus adjacent wordmark or brand-name text is one logo unit
. A truly standalone logo may remain a direct layer item.
- A chart assembled from multiple layers groups its data marks, axes,
labels, and chart-specific container into one chart unit. Separate
charts must not be merged merely because their layers share gv.

Proximity, containment, and decoration rules:
- Nearby layers should normally be grouped when they overlap, touch,
contain one another, share a tight horizontal/vertical alignment, or
have a small gap relative to surrounding spacing.
- Ignore overlap with a full-page mb background when finding local
units. For all other layers, overlap/contact is a strong keep-together
 constraint, not merely optional evidence.
- At any parent, do not leave two overlapping/contacting non-
background items as separate direct siblings when one decorates,
supports, contains, or continues the other. Put them in one local
group or one shared region group.
- At root, an mx/gc/ge/gl/gi/gb layer that overlaps, touches, or is
clearly adjacent to foreground text/media/chart cannot remain an
unrelated direct leaf. Attach it to that foreground unit or to the
smallest region containing both.
- Do not use a large hero image, large container, or region-spanning
layer as a bridge that pulls decorations from different semantic
regions into one group. Overlap chains are valid only when every piece
 clearly serves the same local target.
- Proximity selects the correct semantic partner. Do not pair distant
elements of the same category when a nearby corresponding partner
exists.
- A container/frame/card/panel/button/background shape that contains
or overlaps text, media, chart, or icon content belongs in one group
with that content.
- Multiple information blocks inside one outer container should remain
 separate child groups under the larger container group.
- A major region group is not a substitute for its local groups. When
a region contains distinct heading/body, contact/footer, badge, media-
caption, or repeated-item units, represent those units as nested child
 groups instead of placing all their layers flat in one catch-all
items list.
- If two or more layers already form a clear local semantic unit, keep
 that unit nested when wrapping it in a larger region. Do not dissolve
 the local unit merely to reduce root-level leaves.
- Before emitting a group with five or more direct layer children,
perform a mandatory local-unit pass. If any pair or subset forms a
heading/body, text/decoration, footer/contact, logo/ornament, media/
decoration, badge/text, icon/text, or container/content unit, that
subset must be a nested child group. A large all-leaf group is valid
only when all layers truly form one indivisible local visual unit.
- An icon, gb marker, ge, or gl belongs with the adjacent text/content
 that it labels, points to, selects, marks, or emphasizes.
- Decoration follows information. An mx or ge layer beside, touching,
overlapping, behind, or at the corner of a specific text/media unit
belongs in the same local group as that unit.
- Determine a target independently for every decoration before
building larger regions. Attach it to the nearest semantically
plausible target, considering distance, side, corner placement,
alignment, and visual intent.
- Never collect most or all mx layers into one media/hero group merely
 because they share mx or because a large media bbox spans the page.
Decorations near a text column, heading, footer, logo, or media belong
 to those respective local units.
```

```
- Decorations placed directly below, beside, or aligned with a heading
/body block should remain with that information block even when a
large hero image exists elsewhere on the page.
- When decoration frames or balances an entire information region
rather than one leaf, put it directly inside that region group
alongside the region's child groups.
- Leave mx directly at root only when it is ambient page decoration
attached to no foreground information or media region.
- Nearby pieces that together form one indivisible ornament may form
one local group, but never create a category bucket for all decoration
.
- For repeated cards/items, first create one complete group per card/
item, including its own container, media, text, marker, and decoration
. Then place those item groups under one repeated parent.
- Never group layers only because they share a slot. Distant text,
media, icons, lines, or decoration remain separate unless one visible
semantic structure unifies them.

Avoid excessive ungrouped leaves:
- A direct layer item is valid, but it is a last resort for a
genuinely standalone element, not the default representation.
- Background layers may stay directly under root. Other root-level
leaves require another check for a nearby target, information block,
hero region, footer/header region, or shared container.
- A direct root mx is allowed only when it has no meaningful overlap,
contact, or close attachment to any foreground unit or region.
- Do not leave title, supporting body, and nearby decoration as
separate root children when they visibly form one message block.
- Do not leave hero media, overlapping support shapes, and nearby
ornaments as unrelated root children when they form one hero visual
region.
- Do not leave contact/footer text and its adjacent icon, mark, line,
or decoration as unrelated root children.
- Reduce ungrouped leaves by building real multi-layer semantic groups
 and region hierarchy. Never reduce them by inventing a one-layer
wrapper or grouping unrelated distant content.
- Do not solve the ungrouped-leaf check by putting most foreground
layers into one shallow catch-all group. A large region with multiple
messages or functions needs meaningful child groups.
- A text column containing top metadata, a main heading/body message,
and footer/contact information is a larger region. Preserve a nested
heading/body group and any local footer/decor group inside it instead
of flattening the whole column.

Reading and design order:
- At root and section levels, order semantic regions by intended
reading flow: normally top-to-bottom, and left-to-right among regions
on the same visual band.
- Inside a text/information block, order title, subtitle, body,
caption, labels, and metadata according to how they are read.
- Inside a local contained unit, place its background/container/
support shape before the foreground content it supports, then keep
readable content in reading order.
- Order repeated cards/items left-to-right for a row and top-to-bottom
 for a column; for a grid, process rows from top to bottom and items
within each row from left to right.
- Do not use layer id, paint order, or slot category as a substitute
for visual reading order.

Final checks:
1. Every id from layers_norm1000 occurs exactly once as an integer "
idx".
2. No output id is absent from layers_norm1000.
```

```
3. Every layer item has exactly "idx" and "slot", and its slot is one
of the listed codes.
4. Every group has exactly "items" and at least two descendant layers;
 no group has a "slot".
5. No group exists only because its members share a slot.
6. Every obvious container-content, badge-text, icon-text, marker-text
, line-target, decoration-target, chart, and media-caption unit is
grouped locally.
7. Heading and supporting content that communicate one message are
grouped, and nearby decoration is attached to that group or its region
.
8. Root contains major regions, backgrounds, and only genuinely
standalone leaves; it is not a flat list of avoidable single layers.
9. Every items list follows visual reading/design order.
10. Large region groups preserve their obvious local semantic child
groups rather than flattening all descendant layers into one level.
```

G.4 Prompt for the Composition Quality Reward

```
You are a strict expert evaluator comparing {candidate_count} graphic-
design relayout candidates against one source design.

You receive the SOURCE DESIGN first, followed by anonymous candidates
with IDs {candidate_ids}. Candidate order is randomized. The target
reference layout is intentionally absent because multiple relayouts
may be valid.

Evaluate all candidates jointly so that scores are calibrated on a
shared scale, but judge each candidate against the SOURCE DESIGN
rather than against other candidates. Use only visible evidence. Do
not penalize defects already present in the source.

Major failures.
Strongly penalize missing, duplicated, substituted, or unintentionally
 hidden elements; destructive overlap, cropping, overflow, deformation
, or incorrect stacking; missing, corrupted, clipped, or unreadable
text; loss of important information, visual hierarchy, or reading
order; and layouts that mechanically retain the source arrangement
when it does not fit the target canvas.

Composition quality.
For candidates without major failures, prefer clear visual hierarchy
and reading flow; coherent alignment, grouping, and relative element
sizes; balanced margins, spacing, density, and visual weight;
effective canvas utilization; appropriate cropping, reflow, and
repositioning; and a polished composition that preserves the source
identity.

Score calibration.
90--100: excellent adaptation, with no meaningful new defect.
70--89: good adaptation, with only minor local issues.
50--69: acceptable but noticeably degraded.
25--49: poor adaptation with substantial problems.
0--24: severely broken adaptation.

Use the full score range. Within the same quality band, assign
different scores when visible differences justify it. For otherwise
similar candidates, use hierarchy, alignment, spacing, balance, and
canvas utilization as tie-breakers.
```

```
Return every candidate ID exactly once with a single integer score and
 a concise justification. Return only JSON matching the supplied
schema.
```

G.5 VLM Evaluation Prompt

```
You are a strict expert evaluator of graphic-design relayout quality.

You will receive exactly two images:
1. SOURCE DESIGN: the original layered design on its original canvas.
2. OUTPUT DESIGN: a proposed relayout of the same design on a target
canvas.

The target reference layout is intentionally not provided. Graphic-
design resizing has no single canonical solution. Judge whether the
output is a faithful, usable, and polished adaptation of the source
rather than whether it reproduces particular coordinates. Changes in
arrangement, scale, spacing, or line wrapping are allowed when they
are coherent responses to the new canvas.

Compare only visible evidence in SOURCE and OUTPUT. Penalize defects
introduced or made worse by the output; do not penalize an issue
merely because it already exists in the source. Assign each defect
primarily to the dimension that best captures its cause and effect,
and do not mechanically deduct for the same issue in multiple
dimensions.

Score the output on four dimensions. Use integer scores only.

D1. Element Fidelity (0-20)
Compare OUTPUT with SOURCE and evaluate the visible completeness of
every element together with the visual quality of non-text assets.
- Penalize newly introduced or worsened occlusion, unintended
overprinting or collisions, off-canvas overflow, destructive clipping
or cropping, and elements that visibly burst beyond their intended
bounds.
- Penalize a whole meaningful element that is missing, duplicated,
newly added, substituted, or replaced when this changes the design's
content or visual identity.
- For photographs, illustrations, logos, icons, and other graphic
assets, penalize new blur, rendering artifacts, deformation, incorrect
 aspect ratio, or cropping that removes a key subject or identifying
feature.
- Penalize any visibly broken, incomplete, or unfinished layer state.
- Intentional cropping is acceptable when it is compositionally sound
and preserves the asset's recognizable subject and role.
- Score a completely missing or hidden text layer here; score the
content, typography, and readability of a present text layer under D2.

D2. Text Readability (0-20)
Compare SOURCE and OUTPUT for textual accuracy, typography, and
readability.
- Penalize missing, substituted, corrupted, or mismatched letters,
words, numbers, punctuation, and other key information.
- Penalize line breaks that damage meaning, orphaned characters or
words, isolated single-word lines, clipped glyphs or lines, and
abnormal letter spacing or line spacing that harms reading.
- When glyphs are otherwise intact, penalize newly introduced
difficulty caused by insufficient contrast or interference from the
background.
```

```
- A change in where a line wraps is not by itself a content error or
readability defect when all text remains accurate, complete,
semantically coherent, and easy to read.
- Judge readability relative to the source's intended typography.
Unusual typography that is already present in SOURCE is not an output
defect unless it is worsened.

D3. Relation Preservation (0-30)
Compare SOURCE and OUTPUT and evaluate whether the relationships among
 elements are preserved or sensibly adapted.
- Preserve visual hierarchy and reading order, including the relative
prominence of titles, key messages, hero media, supporting content,
logos, metadata, and decoration.
- Preserve or coherently adapt alignment, relative size, grouping,
containment, adjacency, repetition, relative position, and local
spacing.
- Preserve meaningful overlap, underlay, and front/back relationships
unless changing them is clearly beneficial for the target canvas.
- Exact coordinates, identical rows or columns, and identical gap
sizes are not required. Reflow, scaling, and spacing changes are valid
 when they form a consistent adaptation to the new aspect ratio.
- If most elements in a group follow one clear layout rule but an
individual element visibly deviates from that rule, treat the outlier
as degraded relation preservation.
- Penalize mechanically copying the source arrangement when that fails
 to adapt the relationships coherently to the target canvas.

D4. Composition Quality (0-30)
Evaluate the OUTPUT as a finished full-canvas composition. Compute D4
as the sum of the following three mandatory internal subscores.

A. Canvas utilization and content scale (0-15)
- Inspect how meaningful content occupies the target along both axes,
including its overall span, scale, and distribution across the canvas.
 Core content should make deliberate use of the target shape rather
than remain at source-like scale or collapse into one region.
- A full-canvas background, a large plain container, or isolated
decorative marks do not by themselves count as effective use of space.
 Judge where content-bearing text, media, logos, and meaningful visual
 groups are placed.
- Explicitly inspect the top, bottom, left, right, and center for
contiguous dead bands, unused quadrants, or weakly occupied regions,
especially when another region is crowded.
- Whitespace is valid only when it visibly supports hierarchy,
separation, or the source's intended character. Do not excuse a large
empty region merely because some whitespace existed in SOURCE;
determine whether the OUTPUT has deliberately recomposed the content
for the new aspect ratio.
- Give 13-15 only when content scale and occupied span make convincing
 use of the target with no conspicuous accidental dead region; 9-12
for mild inefficiency; 5-8 for a clear large dead band, underscaled
composition, or one-sided crowding; and 0-4 when the layout collapses
into a small area or roughly half the canvas is functionally unused.
- If a contiguous accidental dead band occupies roughly one quarter or
 more of a target dimension while content is cramped elsewhere, this
subscore must be at most 8 and D4 must be at most 18. If roughly half
the target is functionally unused, this subscore must be at most 4 and
 D4 must be at most 12.

B. Balance, margins, and rhythm (0-8)
- Assess horizontal and vertical visual weight, center-versus-edge
balance, margins, padding, gaps, repeated spacing, and whether dense
and sparse regions form an intentional rhythm.
```

```
- Penalize arbitrary edge crowding, lopsided visual weight,
inconsistent margins, awkward isolated elements, and spacing that
makes the composition feel unstable.

C. Finish and coherence (0-7)
- Assess focal hierarchy, reading flow, aesthetic coherence, and
whether the result looks deliberately composed for the target rather
than mechanically copied, stretched, cropped, or left as a placement
draft.

The D4 explanation must explicitly report the three internal subscores
 in this exact style: 'Utilization X/15; Balance Y/8; Finish Z/7',
followed by concise visible evidence. D4 must equal X + Y + Z. A D4
score of 24 or higher requires Utilization of at least 12 and no
conspicuous accidental dead region.

Overall score calibration:
- Level 5 / 91-100: No newly introduced visible degradation is found;
the adaptation is equivalent in quality and function.
- Level 4 / 66-90: Degradation is local and non-critical; the primary
relationships and overall usability remain intact.
- Level 3 / 41-65: Clear degradation is present and requires
improvement, but the core content remains recognizable and usable.
- Level 2 / 16-40: Critical regions or multiple relationships are
damaged; substantial correction is required.
- Level 1 / 0-15: The core structure, primary subject, functional
elements, or essential information is severely damaged.

Be evidence-based and strict. Do not invent hidden layers, unseen
metadata, or defects that cannot be established from the images.
Explain the most consequential visible evidence for each dimension.
The total score must exactly equal the sum of D1, D2, D3, and D4.
Return only JSON matching the supplied schema.
```