ALL-IN-IMAGE: UNIFIED VISUAL CONDITIONING FOR GRAPHIC DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Graphic design generation combines requirements for content, layout, and typography with visual references whose appearance should be retained. Existing systems commonly encode these conditions through separate language, layout, and reference interfaces. Our observation is that different control objectives can share a unified semantic representation. Various requirements can all be expressed on a single spatial canvas, which describes the desired design without depicting its final appearance. Based on this observation, we introduce All-in-Image, a unified visual conditioning method that represents design conditions with two aligned images. On the semantic image, a global prompt, regional prompts, and text-rendering instructions are drawn together with their target regions, allowing semantic content and spatial extent to be represented in the same image. On the appearance image, subject and glyph references are placed at their target locations to provide appearance and glyph structure. A semantic encoder reads the written instructions and their spatial context, while a pixel encoder extracts features that preserve reference appearance. Their tokens jointly condition a single-stream diffusion transformer, allowing language, spatial regions, and appearance references to enter the generator through image space without a conventional text encoder or numerical bounding-box inputs. Experiments on the CreatiDesign benchmark show this formulation supports spatial control, text rendering, and subject fidelity jointly, ranking first on nine of ten metrics.

1 INTRODUCTION

Graphic design generation requires a model to turn a set of design requirements and visual references into a coherent composition. The requested content must be realized with the intended layout and typography, while reference subjects must retain their distinctive appearance (Zhang et al., 2026). These requirements span global creative direction and detailed requirements for individual elements. We study whether these requirements can be communicated through a unified visual representation that conveys design intent while retaining the visual details needed to realize it.

Existing design generators commonly combine language embeddings, layout representations, and reference features through distinct conditioning mechanisms (Li et al., 2023; Zhang et al., 2025b; Tuo et al., 2024; Zhang et al., 2026). Recent image-native approaches further explore expressing instructions and controls directly in visual inputs (Dalva et al., 2026; Liu et al., 2026; Wang et al., 2026; Zhang et al., 2025a). These methods differ in how language, spatial structure, and reference appearance are represented and fused. *Our observation is that semantic requirements and their spatial scope can be represented together in image space.* A global prompt, regional prompts, and text-rendering instructions can all be rendered on one semantic image, with each condition paired with a bounding box in the same color. The global prompt follows the same rule, using a box that spans the full semantic image. The resulting semantic image represents each instruction together with its location and extent, rather than encoding language and layout through separate interfaces. Such a representation describes the desired design without depicting its final appearance.

Based on this observation, we introduce **All-in-Image**, a method for unified visual conditioning in graphic design (Figure 1). It represents the global prompt, regional prompts, and text-rendering instructions together with their target regions on a semantic image read by an image encoder. An aligned appearance image places subject and glyph references at their target locations, where a pixel

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.

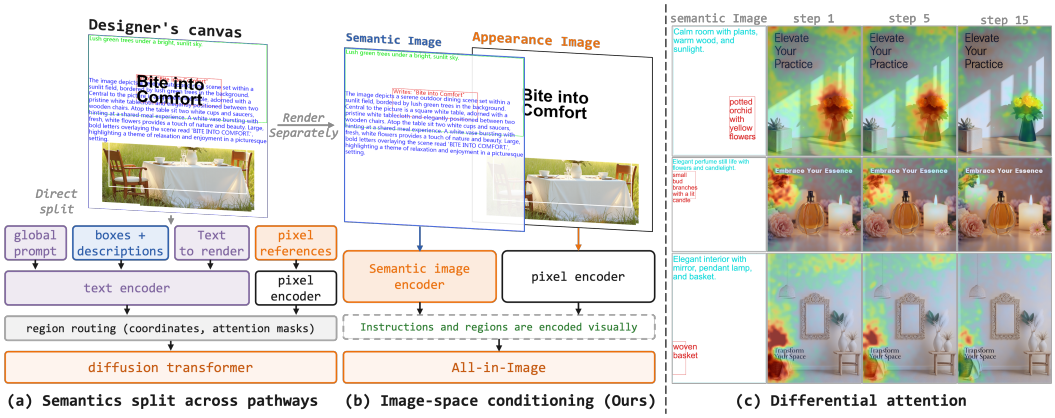


Figure 1: **Conditioning a generator on the requirements of a design.** A designer's canvas is represented by two aligned images: a semantic image containing the global prompt, regional prompts, and text-rendering instructions together with their target regions, and an appearance image carrying subject and glyph references. **(a)** Structured conditioning encodes each requirement through its own interface and assigns it to a region by a routing rule. **(b)** All-in-Image reads the two aligned images with a semantic encoder and a pixel encoder, respectively. **(c)** Attention differences relative to a blank semantic image, shown at three denoising steps for three designs. Responses concentrate around the annotated regions. Computation details are provided in Appendix B.

encoder extracts their appearance features. The semantic, pixel, and noisy target tokens are concatenated and jointly processed by a single-stream diffusion transformer (Z-Image Team et al., 2025). All-in-Image uses image-encoded instructions in place of conventional text-encoder conditioning during generation. Language, spatial regions, and appearance references therefore enter the generator through image space, without numerical bounding-box inputs.

We evaluate this formulation by examining whether the same visual interface can support subject fidelity, text rendering, and regional control jointly. On the CreatiDesign benchmark, our implementation ranks first on nine of ten metrics and in the top two on all ten. Hand-drawn canvases further test the interface beyond the box-and-description format used in training. With the same global prompt, different sketch arrangements produce corresponding changes in the relative placement of the generated subjects, without sketch-specific fine-tuning. Together, the benchmark results and hand-drawn examples support a visual representation that conveys both the requested content and its spatial arrangement across different forms of canvas input. Differential attention concentrates around the annotated regions in the examples of Figure 1(c), providing a complementary view of how the trained generator responds to the semantic image. The main contributions are as follows:

1. We formulate a unified visual conditioning method in which a global prompt, regional prompts, and text-rendering instructions are represented together with their spatial regions on a semantic image, while subject and glyph references are placed on an aligned appearance image. A semantic encoder and a pixel encoder map these images to tokens, which a single-stream diffusion transformer processes without a conventional text encoder.
2. We evaluate how the unified representation supports subject fidelity, text rendering, and spatial control jointly, obtaining the best score on nine of ten benchmark metrics. Hand-drawn canvases demonstrate its use beyond the training annotation format without sketch-specific fine-tuning, while attention analysis examines the spatial responses to rendered region annotations.

2 Related Work

Controllable and Compositional Generation. Controllable image generation commonly introduces separate pathways for different condition types. Spatial control is represented by dense maps or box–phrase pairs (Zhang et al., 2023; Li et al., 2023; Wang et al., 2024), appearance control by reference-image features (Ye et al., 2023), and glyph control by rendered text through a dedicated branch (Tuo et al., 2024). These designs are modular and effective, but semantic content, spatial extent, and reference appearance are encoded through different interfaces and combined by explicit

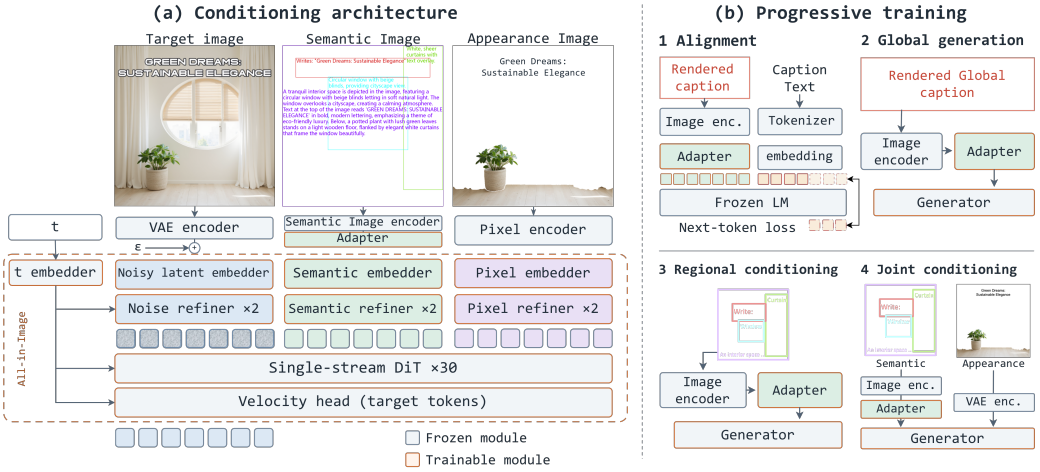


Figure 2: **All-in-Image architecture and progressive training.** **(a)** Separate semantic and pixel encoders prepare the two visual conditions for joint processing with noisy-target tokens in a single-stream DiT. The pixel encoder uses the same frozen VAE encoder as the target path. **(b)** Training proceeds through adapter alignment, global-prompt generation, regional conditioning, and joint semantic–pixel conditioning. The auxiliary language model is used only during alignment. Gray and orange module outlines denote frozen and trainable components, respectively.

routing or fusion rules. All-in-Image represents semantic content and spatial extent on one image, complemented by an aligned appearance image containing subject and glyph references.

Graphic Design Generation. Graphic-design systems extend controllable generation with specialized modules for layout (Zhang et al., 2025b; 2026), typography (Tuo et al., 2024), and subject fidelity (Gao et al., 2025; Cui et al., 2026; Qin et al., 2026). Such specialization enables rich design control, but the resulting systems still rely on condition-specific representations and pathways. Our formulation reduces these heterogeneous inputs to two aligned image-space interfaces.

Visual-Native Conditioning. Closest to our work are methods that move language or control signals into image space. Prompt-Free Diffusion replaces text prompts with visual exemplars (Xu et al., 2024), while UniModel renders prompts as images for visual-only conditioning (Zhang et al., 2025a). Canvas-to-Image places references, poses, boxes, and text annotations on a shared RGB canvas (Dalva et al., 2026), and V2V-Zero uses a visual layout as input to a frozen vision–language model (Liu et al., 2026). UniVL is particularly related: it renders text labels inside masks, reads the composite with an OCR-pretrained encoder, and combines semantic and VAE features according to the mask (Wang et al., 2026). Set-of-Mark prompting further shows that visual marks can expose spatial correspondences to multimodal models (Yang et al., 2023). All-in-Image replaces conventional text-encoder conditioning with two aligned visual inputs for semantic instructions and reference appearance. Their tokens interact with target tokens through full self-attention, with prompt–region associations represented in the image rather than imposed through box-derived routing masks.

3 Method

We represent design conditions as two aligned images, a *semantic image* $\boldsymbol{A}$ and an *appearance image* $\boldsymbol{R}$. The semantic image contains a global prompt, regional prompts, and text-rendering instructions together with their target regions. The appearance image contains subject and glyph references placed at their target locations. Figure 2 summarizes the architecture and training strategy.

3.1 Visual Condition Construction

Let $\boldsymbol{x}$ denote the target design image. We represent each semantic condition by a prompt c_i and its target region b_i. We use $i = 0$ for the global prompt, whose region spans the entire semantic image.

All remaining prompts use their corresponding local regions. For a text-rendering instruction, c_i specifies the target string and its visual attributes, while b_i specifies where the text should appear.

We render all semantic conditions into the semantic image $\boldsymbol{A}$:

$$\boldsymbol{A} = \mathrm{Render}_{\mathrm{sem}}\left(\{(b_i, c_i)\}_{i=0}^{N}\right). \tag{1}$$

Each condition uses a distinct color for both its bounding box and prompt, and the global prompt follows the same rule with a box spanning the full semantic image.

We render appearance references at their target locations in the appearance image $\boldsymbol{R}$:

$$\boldsymbol{R} = \mathrm{Render}_{\mathrm{pix}}\left(\{(b_i, r_i)\}_{i\in\mathcal{I}_{\mathrm{ref}}}\right), \tag{2}$$

where $\mathcal{I}_{\mathrm{ref}}$ indexes elements with subject or glyph references r_i. A subject reference is placed at its target region as either a segmented object or a bounding-box crop. For a text element, a glyph reference is rendered inside the corresponding region.

Object conditions routed to the appearance image are not repeated as regional prompts on the semantic image. A text element can use both images because they provide complementary information: the semantic image specifies the target string and visual attributes, while the appearance image provides glyph structure. Both images use the coordinate system of the target image, so semantic regions and appearance references remain spatially aligned with the desired output. The generator therefore models $p_\theta(\boldsymbol{x} \mid \boldsymbol{A}, \boldsymbol{R})$, with all conditioning information entering through the two visual inputs rather than separate text and coordinate pathways. Accordingly, the architecture requires neither a conventional text encoder nor numerical bounding-box inputs.

3.2 Dual-Encoder Single-Stream Architecture

The two images require different forms of visual encoding. The semantic image requires reading written instructions together with their spatial context, while the appearance image must retain instance-specific appearance. We therefore encode them separately and combine their token sequences inside the diffusion transformer.

Semantic encoding. Image-to-text pretraining supervises the recovery of written content from visual features (Wei et al., 2026), motivating their use as semantic conditions for generation. In place of the base generator's text condition $E_{\mathrm{text}}(c)$, we construct semantic tokens directly from $\boldsymbol{A}$:

$$\mathbf{h}_A = S(P_\phi(E_{\mathrm{img}}(\boldsymbol{A}))), \tag{3}$$

where E_{img} is the frozen image encoder, P_ϕ is a trainable linear adapter, and S comprises the semantic embedder and refiner. Here, c denotes the textual prompt that would enter the base model through its text encoder. The generator learns to use $\mathbf{h}_A$ for both the written requirements and their visual context, without decoding the semantic image back into text or a list of coordinates. The encoder configuration is given in Appendix C.

Pixel encoding. The pixel encoder extracts spatial features for preserving reference appearance and glyph structure. We instantiate it with the frozen VAE encoder used for the target image and take the posterior mean as the reference latent $\boldsymbol{z}_R$. A dedicated pixel embedder and refiner convert its spatial patches into reference tokens $\mathbf{h}_R$, aligned with the target token grid.

Joint generation. We initialize the generator from Z-Image (Z-Image Team et al., 2025) and remove its original text-conditioning path. Given the target latent $\boldsymbol{z}$, Gaussian noise $\boldsymbol{\epsilon}$, and the noise coefficient σ_t, we construct the noisy target latent:

$$\boldsymbol{z}_t = (1 - \sigma_t)\boldsymbol{z} + \sigma_t\boldsymbol{\epsilon}. \tag{4}$$

We convert $\boldsymbol{z}_t$ into target tokens $\mathbf{h}_x$ using the target embedder and refiner, then concatenate the three sequences:

$$\mathbf{h} = [\mathbf{h}_x, \mathbf{h}_A, \mathbf{h}_R]. \tag{5}$$

Full self-attention allows the target tokens to combine semantic instructions and reference appearance at every transformer layer. The two condition inputs remain clean while the target latent is denoised. Target and pixel tokens share two-dimensional coordinates at corresponding locations in the target token grid. Semantic tokens use sequence indices following their order in the image-encoder

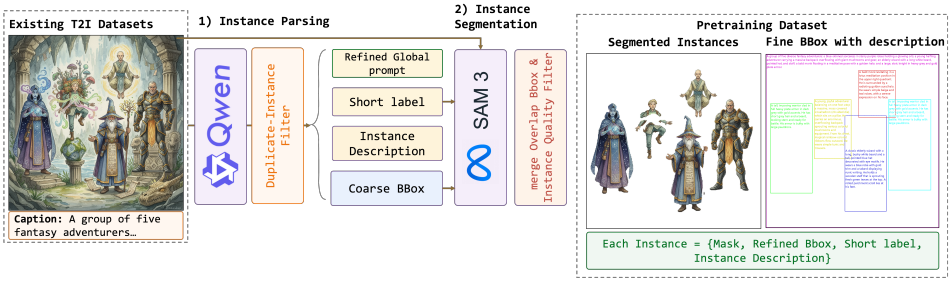


Figure 3: **Structuring image–caption pairs.** A vision–language model locates the entities described in the caption and generates their boxes, labels, and descriptions. After duplicate removal, SAM 3 produces object masks, from which the boxes are recomputed. The resulting annotations support online rendering of descriptions and visual references into the two images.

output. The bounding-box locations and sizes are represented visually in $\boldsymbol{A}$ rather than converted into positional coordinates for the transformer. Exact RoPE indexing is given in Appendix C. Let F_θ denote the diffusion transformer and O the output head. We retain the transformer outputs at the target-token positions and predict the velocity:

$$v_\theta(\boldsymbol{z}_t, t; \boldsymbol{A}, \boldsymbol{R}) = \text{Unpatchify}\left(O(F_\theta(\mathbf{h}, t)_x, t)\right), \tag{6}$$

where the subscript x selects the target-token outputs.

Training objective. We use the rectified-flow objective (Liu et al., 2023b; Lipman et al., 2023) with the data-directed velocity convention specified in Appendix C:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\left[\left\|v_\theta(\boldsymbol{z}_t, t; \boldsymbol{A}, \boldsymbol{R}) - (\boldsymbol{z} - \boldsymbol{\epsilon})\right\|_2^2\right]. \tag{7}$$

The image encoder and VAE remain frozen. Generation training updates the adapter, token embedders and refiners, output head, and diffusion transformer. At inference, $\boldsymbol{A}$ and $\boldsymbol{R}$ remain fixed while a noise latent is iteratively denoised and decoded by the VAE.

3.3 Progressive Training Strategy

Replacing the semantic conditioning path requires the generator to learn a new visual interface before combining multiple forms of design control. We introduce the conditions progressively, as shown in Figure 2(b). Each stage initializes the next.

Semantic encoder alignment. We adopt a standard vision–language feature alignment procedure (Liu et al., 2023a). We render a caption c, with token sequence $y_{1:M}$, on $\boldsymbol{A}_c$ and obtain the visual prefix $\mathbf{V}_c = P_\phi(E_{\text{img}}(\boldsymbol{A}_c))$. We concatenate $\mathbf{V}_c$ with the caption token embeddings and train the adapter through the frozen language model used as the generator's text encoder:

$$\mathcal{L}_{\text{align}}(\phi) = -\mathbb{E}_{(\boldsymbol{A}_c, c)}\left[\sum_{j=1}^{M} \log p_{\text{LM}}\left(y_j \mid \mathbf{V}_c, y_{<j}\right)\right], \tag{8}$$

where each caption token is predicted from the full visual prefix and preceding text tokens. The image encoder and language model remain frozen, and only P_ϕ is updated using text-token prediction targets. The trained adapter initializes subsequent generation training. The auxiliary language model is discarded after adapter alignment and is not used during diffusion training or inference.

Global-prompt generation. We train the adapter and generator on semantic images containing only the global prompt. The diffusion objective now teaches the model to generate the requested content from image-encoded instructions instead of text-encoder features.

Structured semantic conditioning. We add regional prompts, their colored bounding boxes, and text-rendering instructions. Training now requires the model to associate each instruction with its target region and generate the requested content there.

Joint semantic and pixel conditioning. We introduce the appearance image and train the complete model with both visual conditions. This stage combines instruction-driven regional generation with reference-appearance and glyph preservation. Training proceeds to native-aspect high-resolution images and mixes structured designs with synthetic visual-text data.

3.4 Training Data Construction

Existing datasets do not directly provide paired semantic and appearance images. We therefore convert three complementary data sources into the same $(\boldsymbol{A}, \boldsymbol{R}, \boldsymbol{x})$ format. Structured graphic designs provide dense multi-condition supervision, image–caption pairs expand object and semantic diversity, and synthetic visual-text images strengthen exact text rendering.

Structured design data. For CreatiDesign (Zhang et al., 2026), we construct the two images online from its global description, regional annotations, and object masks. Objects used as appearance references are placed on the appearance image, while the remaining objects are represented by regional prompts on the semantic image. Text elements provide text-rendering instructions on the semantic image and optional glyph references on the appearance image.

Image–caption data. We further convert captioned images into regional training conditions. As shown in Figure 3, a vision–language model identifies foreground entities and describes them, and SAM 3 (Carion et al., 2026) provides their masks. The resulting descriptions, regions, and object cutouts are passed to the same renderer used for structured designs.

Synthetic visual-text data. Finally, we render synthetic text images with varied strings, fonts, colors, and layouts. Their semantic images specify the target text and visual attributes, while their appearance images provide aligned glyph references. These samples provide additional supervision for exact text generation. Further construction and sampling details are given in Appendices E–G.

4 Experiments

4.1 Setup

We follow the image-generation and evaluation protocols of CreatiDesign benchmark (Zhang et al., 2026), measuring subject fidelity, layout adherence, and text rendering. The semantic and appearance images are constructed from the benchmark conditions. We compare with FLUX.1-dev and FLUX.1-Fill (Black Forest Labs, 2024), Z-Image (Z-Image Team et al., 2025), UNO (Wu et al., 2025), MS-Diffusion (Wang et al., 2025), CreatiLayout (Zhang et al., 2025b), BizGen (Peng et al., 2025), Emu2 (Sun et al., 2024), CreatiDesign (Zhang et al., 2026), PosterMaker (Gao et al., 2025), and OmniGen (Xiao et al., 2025). Metric definitions, inference settings, and baseline score sources are detailed in Appendix H.

4.2 Main Results

Quantitative Comparison. Table 1 compares All-in-Image with eleven baselines across subject fidelity, layout adherence, and text rendering. All-in-Image ranks first on nine of the ten metrics and second on CLIP-I, making it the only method ranked in the top two across all metrics. Compared with CreatiDesign, the closest baseline in task scope, it improves all ten metrics. Pretrained Z-Image achieves a sentence accuracy of 79.84 but a Text Spatial score of 18.15. Relative to this text-to-image starting point, All-in-Image improves sentence accuracy to 85.60, Text Spatial to 60.12, and layout Spatial from 59.00 to 80.57. Together, these comparisons show that the proposed representation supports the three condition families within one model. Serializing regional descriptions, target strings, and normalized boxes into Z-Image's text prompt without fine-tuning lowers all Layout and Text scores relative to caption-only prompting. The two zero-shot settings are compared in Appendix J. We additionally report joint automatic pass rates on 867 designs with all three annotated condition types. At a mean DINO-I threshold of 0.8, All-in-Image reaches 46.14% versus 32.87% for locally evaluated CreatiDesign under the content-based rule; the definitions, location-aware variant, and threshold sensitivity are reported in Appendix K. Under the inference settings in Appendix M, All-in-Image generates a 1024×1024 image in 41.71 s versus 53.56 s for CreatiDesign, with peak allocated memory of 15.56 versus 36.71 GiB. A supplementary user study compares aesthetic preference and instruction following against CreatiDesign and UNO (Appendix L).

Qualitative Comparison. Figure 4 compares how the methods satisfy subject, text, and regional requirements within the same design. In (a), (b), and (e), All-in-Image retains the reference subjects' appearance, whereas UNO and BizGen alter the food, containers, or product composition. CreatiDesign preserves these subjects more closely, but renders an incorrect title in (a), adds an

Table 1: **Quantitative results.** All-in-Image ranks first on nine of ten metrics and second on CLIP-I on the CreatiDesign benchmark. Baselines are grouped by conditioning type following Zhang et al. (2026). All metrics are higher-is-better, with the global best and second-best highlighted as **best** and second, respectively. Parameter-count sources and conventions are detailed in Appendix M.

Method	Params	Subject ↑			Layout↑				Text ↑		
		CLIP-I	DINO-I	M-DINO	Spatial	Color	Texture	Shape	Spatial	Sen. Acc	NED
Text-to-image											
FLUX.1-dev	16.78B	75.93	44.59	17.76	60.02	47.10	46.19	44.76	13.25	57.95	81.52
Z-Image	10.26B	75.68	44.35	17.64	59.00	49.14	49.95	47.10	18.15	79.84	92.58
Multi-subject image-driven											
UNO	17.83B	77.97	47.88	20.79	53.10	43.44	42.62	41.30	11.47	40.87	74.51
FLUX.1-Fill	16.87B	**90.79**	87.32	69.05	67.55	57.48	56.26	55.75	12.48	12.07	56.69
MS-Diffusion	5.77B	84.75	74.13	44.34	49.54	33.37	34.89	34.79	1.01	0.00	10.21
Semantic layout-driven											
CreatiLayout	8.97B	78.41	55.54	25.31	77.42	63.07	62.67	60.21	18.59	12.27	74.21
BizGen	3.95B	79.86	53.08	22.93	79.84	62.96	62.86	61.01	50.44	75.89	94.61
Multi-condition driven											
Emu2	39.52B	73.96	45.17	19.92	60.81	45.37	46.20	44.06	0.20	0.00	13.81
CreatiDesign	17.36B	89.39	86.48	65.75	78.94	66.02	66.94	65.82	56.90	78.30	94.68
PosterMaker	10.78B	90.45	87.72	69.56	56.36	45.37	44.25	41.61	28.42	0.70	28.62
OmniGen	3.85B	82.15	58.83	30.86	53.92	44.35	44.46	41.40	8.24	6.72	49.12
Ours	7.05B	90.74	**89.81**	**72.76**	**80.57**	**73.04**	**71.82**	**70.80**	**60.12**	**85.60**	**97.15**



Figure 4: **Qualitative comparison on the CreatiDesign benchmark.** Each row shows the semantic and appearance images supplied to All-in-Image, followed by outputs from UNO, BizGen, CreatiDesign, and All-in-Image. Rows (a), (b), and (e) emphasize joint subject and text fidelity, while (c) and (d) emphasize regional instruction following. Red boxes mark text mismatches, purple boxes mark subject differences, and orange boxes mark missing or incorrect regional content.

unrequested line in (b), and repeats "Candy" in (e). All-in-Image renders the requested wording in each of these examples while incorporating the subject into the surrounding scene. The regional instructions also distinguish the outputs. In (c), All-in-Image places a boy in a grey hoodie beside

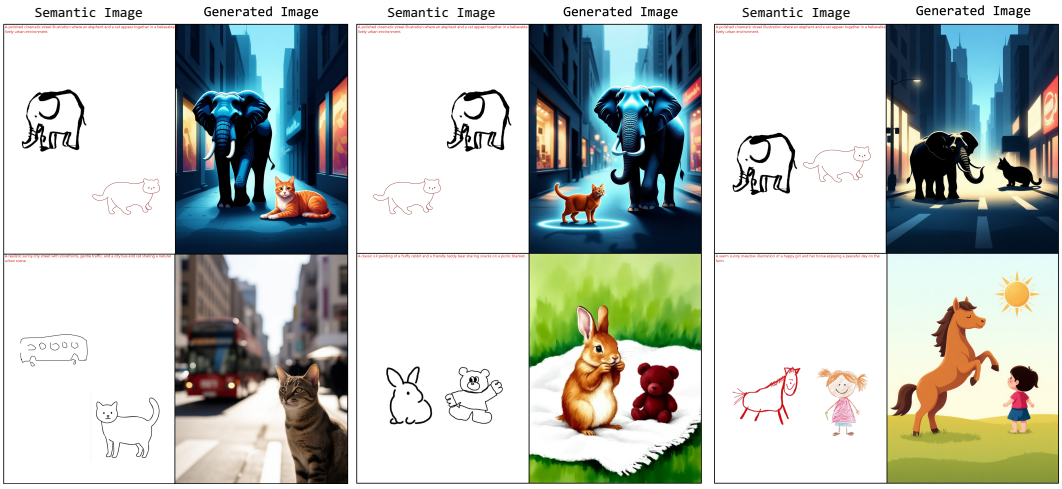


Figure 5: **Zero-shot generation from hand-drawn canvases.** Each pair shows an input canvas and its generated image. **Top:** the same global prompt accompanies three arrangements of an elephant and a cat; the generated subjects follow the sketches' relative placement. **Bottom:** sketches combined with different descriptions. All examples use the model without sketch-specific fine-tuning.

Table 2: **Semantic encoder comparison.** Layout adherence (%, ↑) on 152 square designs at CFG 7. Frozen encoders use image-only self-supervised learning (SSL), global vision–text (V–T) contrastive alignment (align.), or OCR text reconstruction (recon.) as pretraining objectives.

Encoder	**Pretraining**	Spatial	Color	Texture	Shape
DINOv2-L	Vision-only SSL	4.79	1.37	2.05	1.37
CLIP-L/14	V–T coarse align.	21.23	13.70	13.01	11.64
DeepEncoder	V–T recon.	**30.82**	**21.23**	**21.23**	**19.18**

Table 3: **Pixel encoder ablation.** Subject fidelity (%, ↑) with the same appearance references. Shared uses the semantic DeepEncoder; CLIP and VAE use dedicated pixel encoders.

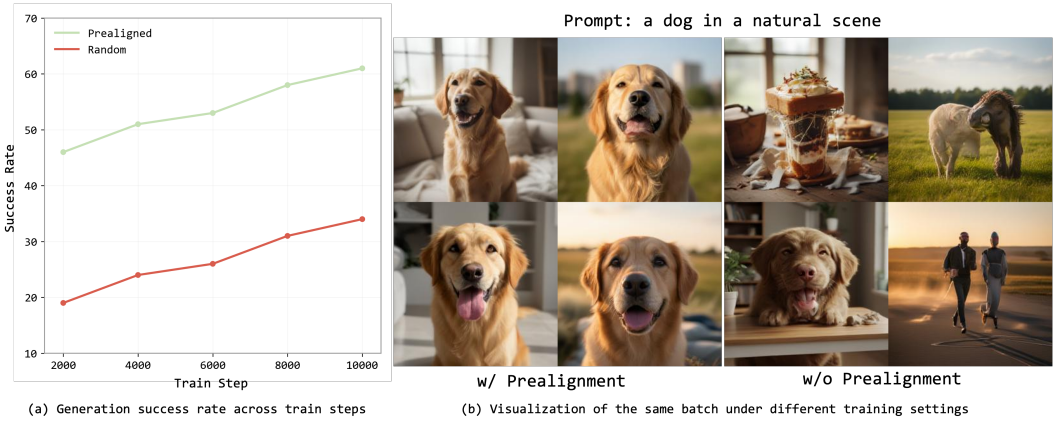
Encoder	CLIP-I	DINO-I	M-DINO
Shared	75.30	37.49	13.90
CLIP-L/14	75.57	38.08	13.87
VAE (Ours)	**90.47**	**88.78**	**71.00**

the reference woman, whereas CreatiDesign renders a red hoodie and UNO and BizGen introduce additional people. In (d), All-in-Image generates the requested white cup and black bowl at the lower left and spells "ModaTap" correctly; CreatiDesign omits both objects and drops an "A" from the name. In (e), All-in-Image includes the upper-left leaves that CreatiDesign omits. Together, these examples show how subject appearance, requested text, and regional object requirements can be satisfied within the same design.

Zero-Shot Generation from Hand-Drawn Canvases. Training represents regional conditions primarily as bounding boxes and textual descriptions. At inference, we also supply semantic images containing hand-drawn sketches and a rendered global prompt, without additional sketch-specific fine-tuning (Figure 5). In the top row, the global prompt is identical and does not specify left–right positions; changing the sketches' arrangement changes the relative placement of the generated elephant and cat. The bottom row illustrates the range of visual styles obtained with different sketches and global descriptions. These examples show that the learned canvas interface extends beyond the box-and-description format used in training, while allowing changes in pose and contour.

5 Ablations

Semantic Encoder Comparison. The semantic image contains both written instructions and spatial annotations, so its encoder must recover linguistic content together with the visual structure in which it appears. We compare three complete pretrained encoders under the same canvas rendering, generator, downstream training schedule, and evaluation set, retaining their native input resolutions and tokenization. Each encoder remains frozen and uses its own trainable adapter. DINOv2-L (Oquab et al., 2023) is trained through image-only self-supervision. CLIP-L/14 (Radford et al., 2021) introduces language supervision through contrastive alignment between global image and text embed-



(a) Generation success rate across train steps

(b) Visualization of the same batch under different training settings

Figure 6: **Effect of semantic-encoder prealignment.** Left: percentage of 100 fixed prompts for which an object detector finds the specified target object in at least 1 of 4 generated images, measured at five training checkpoints. Right: two matched output comparisons under the same conditions.

dings. DeepEncoder V2 (Wei et al., 2026) is trained within an OCR system that predicts text tokens directly from visual inputs, providing a more fine-grained visual-to-text reading signal. Table 2 reports layout adherence on 152 square designs from the CreatiDesign benchmark at CFG 7. Spatial accuracy increases from 4.79 with DINOv2-L to 21.23 with CLIP-L/14 and 30.82 with DeepEncoder V2, with the same ordering on Color, Texture, and Shape. The OCR-trained configuration achieves the highest layout adherence. More details are in Appendix N.

Pixel Encoder Ablation. The appearance image contains subject references at their target locations, so its reader must preserve both their visual detail and position. We compare three ways of encoding the same appearance references. The shared-encoder baseline uses DeepEncoder to read the appearance references as well as the semantic conditions. The other two configurations retain the semantic encoder and use either CLIP-L/14 or a reconstruction-trained VAE as a dedicated pixel encoder. CLIP's image–text contrastive objective learns features for semantic matching but does not train its image encoder to reconstruct the local appearance of a reference. The VAE instead encodes the aligned appearance image into spatial latents, keeping reference features at positions corresponding to the target grid. Table 3 reports subject fidelity through CLIP-I, DINO-I, and M-DINO. The shared-encoder and CLIP configurations give similar scores, whereas the VAE reaches 90.47, 88.78, and 71.00 on the three metrics. This comparison shows the benefit of the spatial VAE pathway for transferring reference appearance in our setup. More details can be found in Appendix O.

Semantic-encoder prealignment. We compare prealigned and randomly initialized adapters during subsequent generation training. At five checkpoints from 2k to 10k steps, we generate four images per prompt on the same 100 prompts. Figure 6 reports the percentage of prompts for which an object detector finds the specified target object in at least one of the four images. Prealignment leads by 27 percentage points at every checkpoint, from 46% versus 19% at 2k steps to 61% versus 34% at 10k steps. This persistent advantage shows that the alignment stage improves early target-object generation through the visual conditioning path. The prealigned model at 2k steps already achieves a higher detection-based success rate than the model with a randomly initialized adapter at 10k steps.

6 Conclusion

We present a unified visual representation for conditioning graphic design generation. All-in-Image expresses design intent on a semantic image and reference appearance on an aligned appearance image, which are read by a semantic encoder and a pixel encoder, respectively. Together, these images convey every condition, including language, through image space without a conventional text encoder. Benchmark results show that this formulation supports subject fidelity, text rendering, and regional control jointly, while the spatial analyses examine responses to rendered annotations. The resulting method allows heterogeneous design requirements to share an image-space interface whose spatial structure remains available to the generator. Studying richer drawn instructions and more complex compositions is a natural next step toward understanding the scope of this representation.

AI USE STATEMENT

We used ChatGPT and Codex to assist with manuscript drafting and revision, literature discovery, discussion of methodological and experimental choices, formulation and presentation of mathematical objectives, and interpretation of experimental results. Generative image tools were used to explore alternative layouts for method diagrams. We used model-generated image–caption data and programmatically rendered visual-text images for training, as described in Section 3.4. Vision–language models were used to construct regional annotations for training data and to evaluate layout adherence, as described in Sections 3.4 and 4.1 and Appendices F and H. The authors take responsibility for the final methods, experiments, scientific claims, and presentation.

ETHICS STATEMENT

All-in-Image is intended to assist graphic design through controllable image generation. The ability to preserve reference appearance and render requested text could also be misused to create misleading advertisements or unauthorized representations of products and people. Use of reference images should respect consent, privacy, and intellectual property rights. Generated designs should undergo human review for factual accuracy, appropriate representation, and suitability for their intended use. The reported subject-fidelity, layout, and text metrics measure adherence to input conditions, not the truthfulness of depicted products or written claims. The images used for our generation training are synthetic. Synthetic data and pretrained components may carry representational biases that affect the people, products, and styles depicted in generated designs.

REPRODUCIBILITY STATEMENT

Section 3 describes the visual condition representation, model architecture, and progressive training objectives. Appendix C specifies the pretrained components and architectural configuration, and Appendix D reports the training schedule and optimizer settings. Appendices E, F, and G describe condition rendering, regional training annotations, and synthetic visual-text generation. Section 4.1 and Appendix H describe the benchmark inputs, inference settings, metric aggregation, and classifier-free guidance. Appendix N details the semantic-encoder comparison, and Appendix M specifies the latency and memory measurements. We will publicly release the model weights and corresponding code.

REFERENCES

Black Forest Labs. FLUX.1, 2024. URL https://github.com/black-forest-labs/flux.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Benlei Cui, Fangao Zeng, Weitao Jiang, Yuwen Zhai, Haiwen Hong, Longtao Huang, Hui Xue, Wenxiang Shang, and Pipei Huang. SIMPLEPOSTER: A simple baseline for product poster generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14854–14863, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Cui_SIMPLEPOSTER_A_SIMPLE_BASELINE_FOR_PRODUCT_POSTER_GENERATION_CVPR_2026_paper.html.

Yusuf Dalva, Gordon Guocheng Qian, Maya Goldenberg, Tsai-Shien Chen, Kfir Aberman, Sergey Tulyakov, Pinar Yanardag, and Kuan-Chieh Jackson Wang. Canvas-to-image: Compositional image generation with multimodal controls. In *ACM SIGGRAPH 2026 Conference Papers*, pp. 1–11. ACM, July 2026. doi: 10.1145/3799902.3811219. URL https://doi.org/10.1145/3799902.3811219.

Yifan Gao, Zihang Lin, Chuanbin Liu, Min Zhou, Tiezheng Ge, Bo Zheng, and Hongtao Xie. PosterMaker: Towards high-quality product poster generation with accurate text rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8083–8093, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Gao_PosterMaker_Towards_High-Quality_Product_Poster_Generation_with_Accurate_Text_Rendering_CVPR_2025_paper.html.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22511–22521, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Li_GLIGEN_Open-Set_Grounded_Text-to-Image_Generation_CVPR_2023_paper.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023b.

Yaofang Liu, Kangning Cui, Meng Chu, Zhaoqing Li, Suiyun Zhang, Jean-Michel Morel, Xiaodong Cun, Haoxuan Che, Rui Liu, and Raymond H. Chan. Beyond text prompts: Visual-to-visual generation as a unified paradigm, 2026. URL https://arxiv.org/abs/2605.12271.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, et al. DINOv2: Learning robust visual features without supervision, 2023. URL https://arxiv.org/abs/2304.07193.

Yuyang Peng, Shishi Xiao, Keming Wu, Qisheng Liao, Bohan Chen, Kevin Lin, Danqing Huang, Ji Li, and Yuhui Yuan. BizGen: Advancing article-level visual text rendering for infographics generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Yuxin Qin, Ke Cao, Haowei Liu, Ao Ma, Fengheng Li, Honghe Zhu, Zheng Zhang, Run Ling, Wei Feng, Xuanhua He, Zhanjie Zhang, Zhen Guo, Haoyi Bian, Jingjing Lv, Junjie Shen, and Ching Law. Innoads-composer: Efficient condition composition for e-commerce poster generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 32988–32999, June 2026.

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 38th International Conference on Machine Learning*, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

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

Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. AnyText: Multilingual visual text generation and editing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ezBH9WE9s2.

Jiayun Wang, Yu Wang, Weijie Gan, Zhenting Wang, and Wei Wei. UniVL: Unified vision-language embedding for spatially grounded contextual image generation, 2026. URL https://arxiv.org/abs/2605.21611.

X. Wang, Siming Fu, Qihan Huang, Wanggui He, and Hao Jiang. MS-Diffusion: Multi-subject zero-shot image personalization with layout guidance. In *International Conference on Learning Representations*, 2025.

Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6232–6242, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wang_InstanceDiffusion_Instance-level_Control_for_Image_Generation_CVPR_2024_paper.html.

Haoran Wei, Yaofeng Sun, and Yukun Li. DeepSeek-OCR 2: Visual causal flow, 2026. URL https://arxiv.org/abs/2601.20552.

Shaojin Wu, Mengqi Huang, Wenxu Wu, Yufeng Cheng, Fei Ding, and Qian He. Less-to-more generalization: Unlocking more controllability by in-context generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18682–18692, October 2025.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. OmniGen: Unified image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13294–13304, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xiao_OmniGen_Unified_Image_Generation_CVPR_2025_paper.html.

Xingqian Xu, Jiayi Guo, Zhangyang Wang, Gao Huang, Irfan Essa, and Humphrey Shi. Prompt-free diffusion: Taking "text" out of text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8682–8692, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Xu_Prompt-Free_Diffusion_Taking_Text_out_of_Text-to-Image_Diffusion_Models_CVPR_2024_paper.html.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V, 2023. URL https://arxiv.org/abs/2310.11441.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2308.06721.

Z-Image Team, Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, Xin Jin, Liangchen Li, Zhen Li, Zhong-Yu Li, David Liu, Dongyang Liu, Qilong Wu, Feng Yu, Zechao Zhan, Chi Zhang, Shifeng Zhang, Ruikai Zhou, and Shilin Zhou. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer, 2025. URL https://arxiv.org/abs/2511.22699.

Chi Zhang, Jiepeng Wang, Youming Wang, Yuanzhi Liang, Xiaoyan Yang, Zuoxin Li, Haibin Huang, and Xuelong Li. UniModel: A visual-only framework for unified multimodal understanding and generation, 2025a. URL https://arxiv.org/abs/2511.16917.

Hui Zhang, Dexiang Hong, Yitong Wang, Jie Shao, Xinglong Wu, Zuxuan Wu, and Yu-Gang Jiang. CreatiLayout: Siamese multimodal diffusion transformer for creative layout-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18487–18497, 2025b. URL https://arxiv.org/abs/2412.03859.

Hui Zhang, Dexiang Hong, Maoke Yang, Yutao Cheng, Zhao Zhang, Weidong Chen, Jie Shao, Xinglong Wu, Zuxuan Wu, and Yu-Gang Jiang. CreatiDesign: A unified multi-conditional diffusion transformer for creative graphic design. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Wtda8HpVp2.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhang_Adding_Conditional_Control_to_Text-to-Image_Diffusion_Models_ICCV_2023_paper.html.

Table 4: **Where All-in-Image sits among visual-native and design-specific conditioning.** *Interface* lists what reaches the generator besides rendered pixels. *Controls* are what the interface can express for full-page graphic design. ✓ present, ✗ absent, ○ partial or restricted.

Method	Interface			Expressible controls			
	User text tokens	Task token	AR reasoning	Global text	Dense regional language	Multiple references	Full-page synthesis
CreatiLayout (Zhang et al., 2025b)	✓	✗	✗	✓	✓	✗	✓
PosterMaker (Gao et al., 2025)	✓	✗	✗	✓	○	✓	✓
CreatiDesign (Zhang et al., 2026)	✓	✗	✗	✓	✓	✓	✓
Prompt-Free Diff. (Xu et al., 2024)	✗	✗	✗	○	✗	✗	✓
UniModel (Zhang et al., 2025a)	✗	✓	✗	✓	✗	✗	✓
Canvas-to-Image (Dalva et al., 2026)	✓	✓	✗	✓	○	✓	○
V2V-Zero (Liu et al., 2026)	○	✗	✓	✓	○	✓	○
UniVL (Wang et al., 2026)	✗	✗	✗	○	○	○	✗
All-in-Image (ours)	✗	✗	✗	✓	✓	✓	✓

A Interfaces of Related Conditioning Methods

Table 4 summarizes the interface and the expressible controls of the design-specific and visual-native systems discussed in Section 2. Entries are transcribed from each paper's own account of its inputs and task scope. Public code or weights were unavailable for the last three visual-native systems at the time of writing, so the main results do not include them.

B Spatial Responses to Canvas Annotations

We visualize changes in attention from target tokens to semantic image tokens relative to a blank semantic input. Let $a_{\ell,t}(q; \boldsymbol{A})$ denote the attention mass from target token q to the semantic tokens at transformer layer ℓ and denoising step t. We subtract the corresponding attention under a blank semantic image $\boldsymbol{A}_\varnothing$ and average over a fixed set $\mathcal{L}$ of five transformer layers:

$$\Delta a_t(q) = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \left[a_{\ell,t}(q; \boldsymbol{A}) - a_{\ell,t}(q; \boldsymbol{A}_\varnothing) \right]. \tag{9}$$

The attention mass is summed over the semantic-token sequence, so each map visualizes the spatial response to the semantic input as a whole. The resulting scores are arranged according to the spatial positions of the target tokens and visualized over the generated image. Figure 1(c) shows these responses at selected denoising steps. Responses concentrate around annotated regions.

C Architecture Details

Semantic encoder configuration. The main model reported in Section 4.2 uses DeepEncoder V2 (Wei et al., 2026) as its semantic image encoder. It consists of a SAM visual tower, a Qwen2 decoder used as an encoder, and a projector that produces 1280-dimensional features. These components and the pretrained view separator remain frozen. The trainable linear adapter maps the features to 2560 dimensions before the condition embedder maps them to the transformer width. We use the encoder's global and local crop mode. The semantic image is resized to fit within 1024^2 pixels and padded to a square with gray borders for the global view. The local views use two to six crops of 768^2 pixels, with their grid selected according to the input aspect ratio. Local features precede global features, with a learned separator between views. The global view produces 256 tokens and each local crop produces 144, yielding 544–1,120 tokens before separators and padding.

Backbone and pixel pathway. The Z-Image transformer (Z-Image Team et al., 2025) has a width of 3840 and 30 layers, each with 30 attention heads of dimension 128 and QK normalization. The three RoPE axes use dimensions $[32, 48, 48]$. The target and reference share a frozen VAE with 16 latent channels and spatial downsampling by a factor of eight. Target latents are sampled from the posterior during training, whereas reference latents use its mean, with both applying the VAE scale

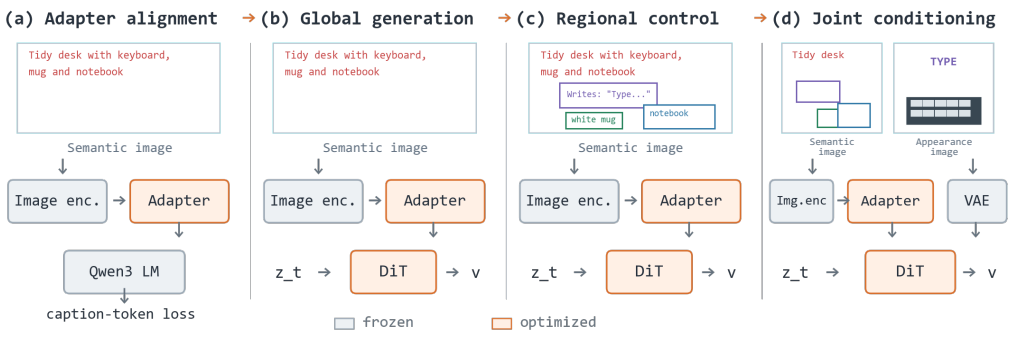


Figure 7: **Progressive training of image-space conditioning.** (a) Caption-token loss through a frozen Qwen3 LM trains only the adapter; the LM is discarded before generation. (b) Global-prompt semantic images train generation before localized instructions are added. (c) Colored boxes, regional descriptions, and visual-text instructions are introduced. (d) An appearance image adds aligned subject and glyph references through a frozen VAE. Gray modules are frozen and orange modules are optimized.

and shift. Both latent streams use 2×2 patches, each containing 64 scalar values before embedding. Each stream has its own embedder and two refiner blocks, and the pixel refiner uses no timestep modulation. The pixel embedder is initialized from the target embedder, and its refiner is initialized from the semantic refiner. Generation training updates the entire transformer and adapter.

Position and time conventions. Let L be the padded semantic sequence length, i a zero-based semantic token index, and (h, w) a location in the target token grid. The three-axis RoPE coordinates are

$$\begin{aligned} \mathbf{p}_{A,i} &= (i+1, 0, 0), \\ \mathbf{p}_{x,h,w} &= (L+1, h, w), \qquad \mathbf{p}_{R,h,w} = (s_{\text{ref}}, h, w), \end{aligned} \tag{10}$$

where $s_{\text{ref}} = 290$ is fixed and is independent of the diffusion timestep. Semantic tokens retain sequence positions rather than being assigned the bounding-box coordinates of individual elements. The target and pixel streams share the same spatial token-grid scale. At 1024×1024 image resolution, a VAE downsampling factor of eight and 2×2 latent patches yield a 64×64 token grid. When larger reference patches are used, their spatial coordinates are mapped to the target token-grid scale. RoPE acts on queries and keys, leaving values unchanged.

For Equation 4, a scheduler index k is sampled uniformly, with noise coefficient σ_k and scheduler timestep τ_k. The model time is $t = 1 - \tau_k/1000$, and σ_t denotes the associated σ_k. Equation 7 uses the data-minus-noise target $\boldsymbol{z} - \boldsymbol{\epsilon}$ under this convention.

D Training Details

For training data and sampling, we use 393,738 images from the CreatiDesign training set (Zhang et al., 2026), an internal corpus of 1,137,476 image–caption pairs, and online-synthesized visual-text images. The internal corpus is processed with the regional annotation pipeline in Appendix F to obtain entity descriptions, bounding boxes, and object masks. Synthetic text images are rendered on the fly with varied strings, fonts, colors, and layouts, as described in Appendix G, rather than stored as a fixed collection.

The internal image–caption corpus is used mainly during early training at 512^2 and lower target resolutions. After switching to multi-resolution bucketing, training draws primarily on CreatiDesign and online-synthesized text images. During mixed-conditioning training, structured and global-prompt-only samples are drawn at a $7:3$ ratio. The stage-specific training schedule is given in Table 5.

Table 5 groups the training sequence into the four stages of Section 3.3, retaining separate rows when configurations change. The recorded final-stage configuration uses AdamW with $\beta = (0.9, 0.999)$,

Table 5: **Training schedule.** Stages correspond to Section 3.3. Image–caption pairs provide global descriptions; structured data includes regional annotations and rendered synthetic text images (Section 3.4). Multiple rows within a stage retain successive configurations. Steps count updates within each configuration, and "–" denotes an unavailable value.

Stage	Data	Target	Steps	GPUs	LR
(a) Alignment	Image–caption pairs	–	1k	8	10^{-5}
(b) Global prompt-only generation	Image–caption pairs	256^2	240k	8	10^{-4}
	Image–caption pairs	256^2	4k	8	10^{-4}
(c) Structured semantic conditioning	Structured data	256^2	30k	16	10^{-4}
	Structured data	512^2	50k	16	10^{-4}
(d) Joint conditioning	Structured data	512^2	100k	8	10^{-4}
	Structured data	Multi-resolution	140k	8	10^{-4}

$\epsilon = 10^{-8}$, weight decay 0.01, a constant learning rate of 10^{-4}, gradient clipping at 1.0, bf16 with TF32 matmuls, gradient checkpointing, and seed 42. Targets are bucketed by native aspect ratio at 1024^2 area. Synthetic slots are re-rendered on every pass with fresh fonts, colors, layouts, and objects, so an epoch is a pass over slots rather than over fixed images.

The Global prompt-only stage begins with the semantic encoder's global view and then adds local crops at 256^2 target resolution. The structured-canvas stage trains at 256^2 and then 512^2 using CreatiDesign and structured image–caption data. The pixel pathway is introduced at 512^2, followed by native-aspect buckets.

E Canvas Rendering

The semantic image keeps the aspect ratio of the target bucket with its long side at 1536 px and a 24 px font. The global text takes the first free full-width band on the semantic image and is narrowed and wrapped only when the image is crowded. Each region is drawn as a complete rectangle in a distinct color; boxes drawn later never erase the borders of earlier ones. A region's description is typeset in the color of its rectangle, placed inside or immediately beside it, and may overflow it. A `Writes` string is given priority over descriptions and borders when they compete for space. Object elements routed to the appearance image are not drawn on the semantic image. Glyphs on the appearance image are set in a simple font and fitted to the box by searching over size, line breaks, line spacing, and rotation; a string that cannot be fitted is skipped and the appearance image keeps its objects. The fit guarantees only that the string fits its container. Per-word size, line assignment, and orientation are not supervised, and 90- and 270-degree candidates cannot be told apart when the source carries no orientation label.

For CreatiDesign, each eligible object is selected as an appearance reference with probability 0.5, with at least one reference selected whenever eligible objects are available. A selected reference uses a bounding-box crop with probability 0.4 and a segmented cutout otherwise. The remaining objects are rendered as descriptions and boxes on the semantic image. Text-reference mode is sampled once per training image. For 70% of samples, the appearance image uses glyphs rendered in a simple font fitted to the text boxes. For 20%, it uses text crops from the synthetic training targets, and for 10%, no additional text reference is included. At evaluation, glyph references are rendered from the provided strings and boxes without using target-image text crops. Object references are retained in all three modes.

F Structuring Image–Caption Data

Figure 3 in the main text shows the offline pipeline. A vision–language model reads the caption of a generated image and grounds the foreground entities it names to coarse boxes with short labels, dropping background regions such as sky or walls. Boxes are clipped to the image, boxes with

Table 6: **Edit operations for synthetic target strings.** Relative sampling weights used to construct edited strings. Each edited string becomes the target of its own training sample.

Operation	Weight	Operation	Weight
Character deletion	160	Character insertion	32
Word deletion	143	Word repetition	5
Character repetition	45	Word insertion	5
Character swap	20	Combined edits	192
Character substitution	63		

a side under 32 px are removed, near-duplicates are merged at IoU 0.6, and at most five entities are kept per image. SAM 3 is prompted with each box at 1024^2 input and returns a mask and a confidence. Masks are mapped back to the original image, boxes are recomputed from the masks, and the caption's description of each entity is attached to its instance. Images are kept when they hold one to five instances, every mask has confidence at least 0.7 and area at least 100 px, and the image is not over-compressed. Of 1,179,510 images entering assembly, 1,137,476 pass. The result is a set of instances, each a mask, a box, a short label, and a description, from which canvases are rendered online exactly as for the structured design data.

G Synthetic Text Images

We construct synthetic training images by rendering sampled strings on solid backgrounds with varied aspect ratios, layouts, fonts, and colors. The semantic image contains background and text-style descriptions, together with a `Writes` instruction at the target box. The appearance image contains aligned glyph references, rendered in a simple font for 80% of samples and in the target font for the remaining 20%. Each sample uses the same string in the target image, text-rendering instruction, and glyph reference.

Half of the samples use unchanged source strings, while the other half use edited strings or strings previously rendered incorrectly by the model. For 95% of primary targets, the normalized string length is less than 30 characters. Table 6 lists the relative sampling weights for character- and word-level edits. In 10% of the edited slots, we reuse strings previously rendered incorrectly by the model.

When the fitted target font is smaller than 36 px, bold upright fonts are preferred. We attempt to add one or two objects with transparent backgrounds to each image, excluding text-like and background layers. If no valid placement is found, the sample falls back to a text-only image.

H Evaluation Details

Benchmark inputs are generated once from the public conditions into a fixed manifest and reused for every All-in-Image run. The appearance image of a benchmark design contains the subject cutouts provided by the benchmark and glyphs fitted from each string and its box. Text metrics are computed per image and averaged (image macro); Sen. Acc counts a string as correct after normalizing case, punctuation, and whitespace. Layout metrics count positive answers over the 983 annotated regions. Subject metrics are computed on all aspect ratios. The main All-in-Image evaluation uses 50 denoising steps, guidance scale $s = 4$, and seed 42 at the benchmark's per-instance output dimensions. In Table 1, Z-Image is evaluated locally, while the remaining baseline performance scores are taken from CreatiDesign (Zhang et al., 2026).

I Classifier-Free Guidance

Classifier-free guidance varies the semantic input while retaining the appearance image. Let $\boldsymbol{A}_\varnothing$ denote a blank semantic image, and define the two predictions as

$$\begin{aligned} p_t &= v_\theta(\boldsymbol{z}_t, t; \boldsymbol{A}, \boldsymbol{R}), \\ n_t &= v_\theta(\boldsymbol{z}_t, t; \boldsymbol{A}_\varnothing, \boldsymbol{R}). \end{aligned} \tag{11}$$

Both branches use the same reference latent and positional encoding. Following the Z-Image pipeline, we combine their predictions as

$$\widehat{v}_t = p_t + s(p_t - n_t). \tag{12}$$

The main results use $s = 4$, giving $\widehat{v}_t = 5p_t - 4n_t$. Under the alternative convention $n_t + w(p_t - n_t)$, this setting corresponds to $w = 5$. The pipeline negates the combined prediction before the scheduler update to match its velocity convention.

We concatenate the positive and blank-semantic branches along the batch dimension and evaluate them in one transformer call per denoising iteration. An output batch size of one therefore uses a transformer batch size of two. Guidance remains active throughout sampling, with `cfg_normalization=False` and `cfg_truncation=1.0`.

J Additional Z-Image Baselines

Global-caption baseline. We evaluate the original, non-distilled Z-Image base checkpoint (Z-Image Team et al., 2025) with its native Qwen3 text encoder using the Diffusers `ZImagePipeline` in bfloat16. Each of the 1,000 benchmark instances uses its global caption verbatim, which may already contain target text. No regional descriptions, box coordinates, text lists, reference images, or glyphs are added. We generate one image per instance using an empty negative prompt, 50 denoising steps, `guidance_scale=4.0`, `cfg_normalization=False`. A CUDA generator is independently reset to seed 42 for each image. Output dimensions follow the same per-instance size manifest used for All-in-Image. The native guidance rule is $p + 4(p - n)$, where p and n are the predictions conditioned on the global caption and the empty negative prompt, respectively.

Structured-prompt baseline. We also evaluate the native Z-Image base model with Qwen3 by serializing the global caption, regional descriptions, and text-rendering instructions into its text prompt. Each regional description and target string is accompanied by its bounding box in normalized $xyxy$ coordinates, with target strings introduced by `Writes`. This setting uses no task-specific fine-tuning, adapter, reference images, or glyphs. We generate all 1,000 designs with guidance scale 4, 50 denoising steps, seed 42, and the same per-instance output dimensions as the main evaluation.

Scoring and results. Both zero-shot settings use the same local Subject, Layout, and Text scoring implementation as All-in-Image. Subject scores are averaged per image over instances with subject references, and Text scores use image-level macro averaging with PaddleOCR2Pytorch. Layout scores count affirmative region-level VQA answers over all evaluated boxes; attributes absent from a description receive the scorer's default affirmative answer. Table 7 compares the two prompt formats on all 1,000 designs. Structured prompting increases DINO-I by 2.38 percentage points and M-DINO by 1.74, while CLIP-I is unchanged at the reported precision. All Layout and Text scores decrease, including Text Spatial by 2.45 percentage points and sentence accuracy by 3.42. Thus, adding regional descriptions, target strings, and coordinates through this zero-shot text serialization does not improve layout or text control. This comparison evaluates prompt formats for the pretrained model. All-in-Image additionally uses task-specific training and appearance references.

K Joint Automatic Pass Rates

We measure whether subject, layout, and text conditions pass their automatic checks within the same generated design. The fixed subset $\mathcal{D}_\cap$ contains 867 benchmark designs with all three annotated condition types. Membership is determined from the input annotations, and the remaining 133 designs are excluded because they lack the object boxes required for Layout evaluation. This analysis uses paired, locally evaluated outputs from All-in-Image and CreatiDesign, rather than the published aggregate baseline scores reported in Table 1.

Content-based rule. Let d_i denote the mean DINO-I score over the subject references in image i, before conversion to a percentage. Let $L_i \in \{0, 1\}$ indicate whether every evaluated object passes all four Layout judgments: Spatial, Color, Texture, and Shape. The existing Layout scorer treats

Table 7: **Z-Image zero-shot prompt comparison.** Both settings use the native text-conditioning path on all 1,000 designs, with guidance scale 4, 50 steps, seed 42, and matching output dimensions. Neither uses reference images or task-specific fine-tuning. Scores are percentages (↑); differences are computed from the displayed scores in percentage points.

Family	Metric	Caption-only	Structured	Difference
Subject	CLIP-I	75.68	75.68	0.00
	DINO-I	44.35	46.73	+2.38
	M-DINO	17.64	19.38	+1.74
Layout	Spatial	59.00	57.07	−1.93
	Color	49.14	47.00	−2.14
	Texture	49.95	47.00	−2.95
	Shape	47.10	43.85	−3.25
Text	Spatial	18.15	15.70	−2.45
	Sen. Acc	79.84	76.42	−3.42
	NED	92.58	92.00	−0.58

Table 8: **Joint automatic pass rates at $\tau = 0.8$.** Both methods are evaluated locally on the same 867 designs selected by their input annotations. Entries give passing counts and percentages; the difference and its paired bootstrap 95% confidence interval are in percentage points (pp). The location-aware rule additionally requires every target text instance to have a scorer-reported single-box IoU of at least 0.5.

Rule	All-in-Image	CreatiDesign (local)	Difference (pp)	95% CI (pp)
Content	400/867 (46.14%)	285/867 (32.87%)	+13.26	[9.34, 17.07]
Content + location	202/867 (23.30%)	149/867 (17.19%)	+6.11	[3.23, 9.00]

unspecified attributes as passing. Let $T_i \in \{0, 1\}$ indicate whether every target string passes the existing OCR content check, which normalizes case, punctuation, and whitespace and applies full-image substring matching. The joint automatic pass rate is

$$J_\tau = \frac{100}{|\mathcal{D}_\cap|} \sum_{i \in \mathcal{D}_\cap} \mathbf{1}[d_i \geq \tau]\, L_i\, T_i, \qquad |\mathcal{D}_\cap| = 867. \tag{13}$$

The subject condition applies to the image-level mean, so individual subjects need not each exceed τ. The text condition checks the presence of the requested strings, without excluding all additional text or requiring a match within each annotated text region.

Location-aware rule. We additionally require every target text instance to pass the scorer's spatial check. Let $\mathcal{T}_i$ be the set of target text instances in design i, and let u_{ir} be the IoU reported by the existing scorer for target instance r and a single OCR detection box. This value is computed from the individual detection box, without merging multiline OCR boxes. The location-aware joint rate is

$$J_\tau^{\text{loc}} = \frac{100}{|\mathcal{D}_\cap|} \sum_{i \in \mathcal{D}_\cap} \mathbf{1}[d_i \geq \tau]\, L_i\, T_i \prod_{r \in \mathcal{T}_i} \mathbf{1}[u_{ir} \geq 0.5]. \tag{14}$$

Both rules combine the existing automatic scores and have not been calibrated against human judgments of complete design success.

Results and threshold sensitivity. We use $\tau = 0.8$ as an exploratory operating point. Under the content-based rule, All-in-Image passes 400 of 867 designs, compared with 285 for CreatiDesign, giving rates of 46.14% and 32.87%. Adding the text-location requirement gives 202 and 149 passing designs, or 23.30% and 17.19%. The differences are 13.26 and 6.11 percentage points, computed from the original counts before rounding. All-in-Image has higher pass rates under both rules at every evaluated threshold from 0.70 to 0.90. Table 9 shows higher joint automatic pass rates for All-in-Image under both rules at all five thresholds from 0.70 to 0.90.

Table 9: **Sensitivity of joint automatic pass rates to the subject threshold.** Rates (%) are evaluated on the same 867 designs. τ thresholds the image-level mean DINO-I, with the Layout and Text rules unchanged. CreatiDesign is evaluated locally. Differences (Δ) are in percentage points, computed before rounding.

	Content			Content + location		
τ	All-in-Image	CreatiDesign	Δ	All-in-Image	CreatiDesign	Δ
0.70	50.29	37.95	+12.34	25.72	19.72	+6.00
0.75	49.37	36.45	+12.92	25.26	18.92	+6.34
0.80	46.14	32.87	+13.26	23.30	17.19	+6.11
0.85	41.18	26.41	+14.76	20.65	13.49	+7.15
0.90	32.76	15.69	+17.07	15.92	8.65	+7.27

Table 10: **Pairwise human evaluation.** W/T/L count judgments favoring All-in-Image, a tie, or the baseline. U denotes unable-to-judge responses, excluded from $N = W + T + L$. Win is $100W/N$, and Pref. is $100(W + T/2)/N$. Raters and prompts denote per-comparison coverage.

Baseline	W/T/L	U	N	Raters	Prompts	Win (%)	Pref. (%)
Aesthetic preference							
CreatiDesign	74/15/42	5	131	23	36	56.5	**62.2**
UNO	68/8/57	5	133	24	35	51.1	**54.1**
Instruction following							
CreatiDesign	78/25/29	4	132	23	36	59.1	**68.6**
UNO	103/4/26	4	133	23	35	77.4	**78.9**

Paired uncertainty estimates. The 95% confidence intervals in Table 8 use paired bootstrap resampling over designs, keeping the two methods' outputs for each sampled design together. The intervals describe variation in the joint-rate differences when resampling images under the fixed rules and existing generated outputs. They exclude uncertainty from new generation seeds, OCR or vision–language model measurement errors, and threshold selection.

Local and published text scores. Locally evaluated CreatiDesign has an image-macro Sen. Acc of 79.62, compared with the published 78.30 used in Table 1. Under the local evaluation, All-in-Image's 85.60 exceeds CreatiDesign by 5.98 percentage points. Requiring all target strings in an image to pass yields 726/1,000, or 72.6%, for CreatiDesign, which is a different aggregation from image-macro Sen. Acc.

L User Study

Setup. We compare All-in-Image with CreatiDesign and UNO through a pairwise user study. The image pool contains 37 prompts with outputs from all three methods. We collected 24 questionnaire responses. Each questionnaire contains 12 comparisons, and each comparison is annotated with two questions: aesthetic preference and instruction following. In each comparison, the left–right order of the two candidate images is randomized and the method names are hidden. Participants are shown the design requirements, the region-and-text requirements, and the subject and glyph references, as illustrated in Figure 8. Responses allow preference for either image, a tie, or inability to judge.

Scoring. We aggregate judgments separately for each criterion and baseline. Let W, T, and L denote the numbers of judgments favoring All-in-Image, indicating a tie, and favoring the baseline, respectively. We report the tie-adjusted preference rate:

$$P = 100\,\frac{W + T/2}{W + T + L}. \tag{15}$$

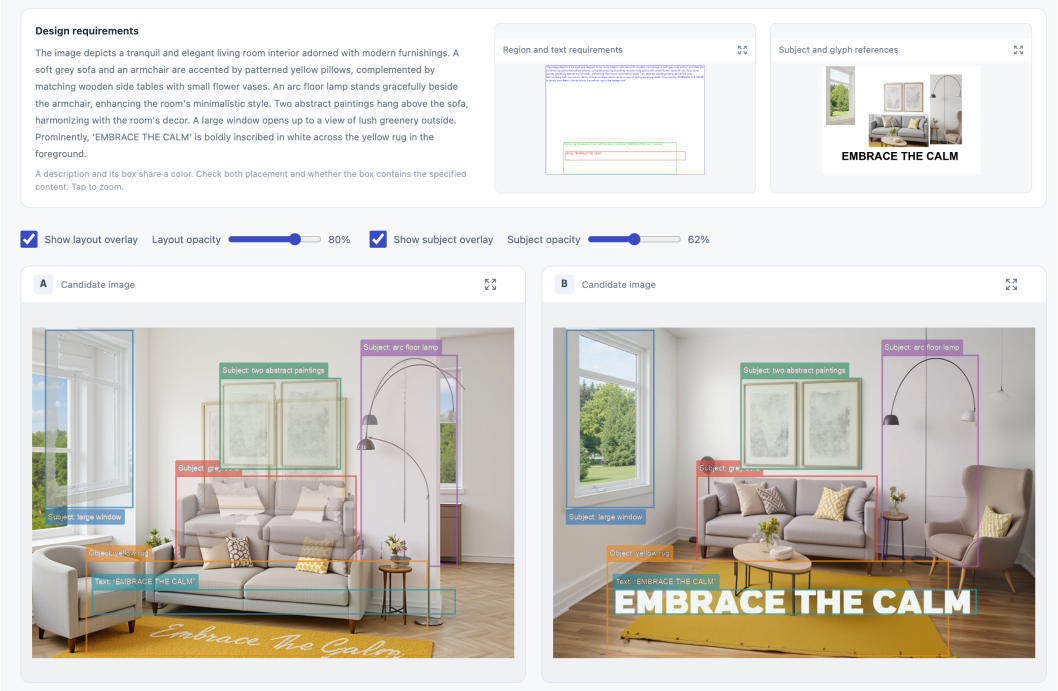


Figure 8: **User-study interface.** Participants are shown the design requirements, the region-and-text requirements, and the subject and glyph references, together with two candidate images. The left–right order of the candidates is randomized and the method names are hidden.

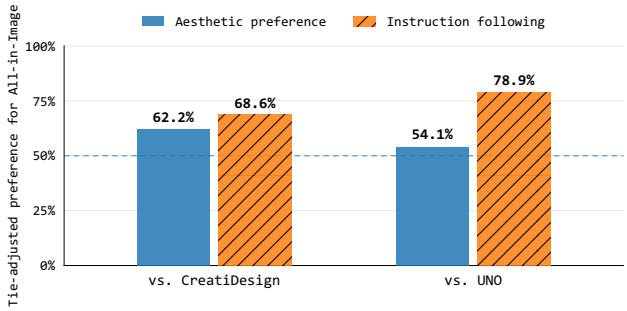


Figure 9: **Pairwise user-study results.** We compare All-in-Image against CreatiDesign and UNO on aesthetic preference and instruction following. Bars report tie-adjusted preference rates, where a tie contributes half a vote to each method.

Unable-to-judge responses are excluded from the denominator and reported separately. The analyzed summary contains 264 aesthetic judgments and 265 instruction-following judgments, after excluding 10 and 8 unable-to-judge responses, respectively. Table 10 provides the counts and coverage for each comparison.

Results. All-in-Image receives tie-adjusted aesthetic preference rates of 62.2% against CreatiDesign and 54.1% against UNO. The corresponding instruction-following rates are 68.6% and 78.9% (Figure 9). The observed preference margins are larger for instruction following than for aesthetics, especially against UNO. These vote-level summaries describe the evaluated respondents and prompt pool.

Table 11: **Inference efficiency.** Both methods generate 1024×1024 images on the same GPU with output batch size one and 50 denoising iterations. Latency is the mean $\pm$ sample standard deviation across eight conditions after averaging two seeds per condition. Memory is the maximum PyTorch allocated or reserved memory over all 16 requests, in GiB.

Method	Resident params (B)	Step (ms) $\downarrow$	Image (s) $\downarrow$	Allocated (GiB) $\downarrow$	Reserved (GiB) $\downarrow$
CreatiDesign	17.362	1064.7 ± 0.6	53.56 ± 0.04	36.71	37.18
All-in-Image	7.051	826.0 ± 0.3	41.71 ± 0.02	15.56	16.02

M Inference Efficiency

We compare All-in-Image and CreatiDesign on the same NVIDIA H800 GPU at 1024×1024 resolution, with output batch size one and 50 denoising iterations. Eight square benchmark conditions are selected at evenly spaced ranks after sorting by the number of layout regions, and each is evaluated with seeds 42 and 43. Each method runs in a separate process after two complete warm-up generations. All required inference components remain on the GPU, with quantization, CPU offloading, compilation, and cross-request condition caching disabled. The environment uses PyTorch 2.6.0, CUDA 12.4, Diffusers 0.36.0, Transformers 5.0.0, and four CPU intra-op threads. Both diffusion transformers use BF16, while the CreatiDesign text encoder retains the FP32 components produced by the installed loader.

Measurement. Step latency is measured with CUDA events from the first transformer forward pass to the end of the scheduler update, including both guidance branches and their combination. We average the measured step latencies within each request after excluding the first two iterations. Image latency is measured with synchronized wall-clock timing from in-memory inputs through condition encoding, all 50 denoising iterations, and VAE decoding to the output image. Model loading, disk I/O, and canvas rendering are excluded. For both latency measures, we first average the two seeds within each condition, then report the mean and sample standard deviation across the eight conditions. Before each measured request, we release unused allocator cache and reset the peak-memory statistics. The reported peaks include resident model weights and inference activations, and take the maximum over all 16 requests. Allocated and reserved memory refer to the PyTorch allocator, with $1\,\mathrm{GiB} = 2^{30}$ bytes. Resident parameter counts include every required inference module, including the VAE and the frozen semantic-encoder view separator.

Conditioning and guidance. All-in-Image uses the main checkpoint with $s = 4$, a semantic image rendered at 1536×1536, and a 1024×1024 appearance image with 4,096 reference tokens. Its two guidance branches are evaluated together, as described in Appendix I. CreatiDesign uses its official conditioning path with a 512×512 subject composite and mask, yielding 1,024 reference tokens. Its positive and negative branches run sequentially with `true_cfg_scale=3.5` and embedding guidance set to one. All-in-Image uses variable-length FlashAttention, while CreatiDesign uses PyTorch SDPA with its structured attention mask. The timings therefore cover each implementation's complete conditioning and sampling path. The native All-in-Image schedule executes 50 transformer calls, including a terminal zero-length update counted in both latency measures.

Results. Table 11 shows that All-in-Image generates an image in 41.71 s, compared with 53.56 s for CreatiDesign, giving a $1.28\times$ speedup. Peak allocated memory decreases from 36.71 to 15.56 GiB, a reduction of 57.6%. The measured denoising-step latency is 826.0 ms for All-in-Image and 1064.7 ms for CreatiDesign. Including the first two iterations in the step average gives 826.0 and 1065.0 ms, respectively.

N Semantic Encoder Ablation Details

The three ablation arms retain their native visual tokenization and use a separate trainable adapter to map frozen encoder features into the generator's conditioning space. DINOv2-L reads a 518^2 input as 1,370 tokens, while CLIP-L/14 reads a 224^2 input as 257 tokens. For the CLIP arm, we use the

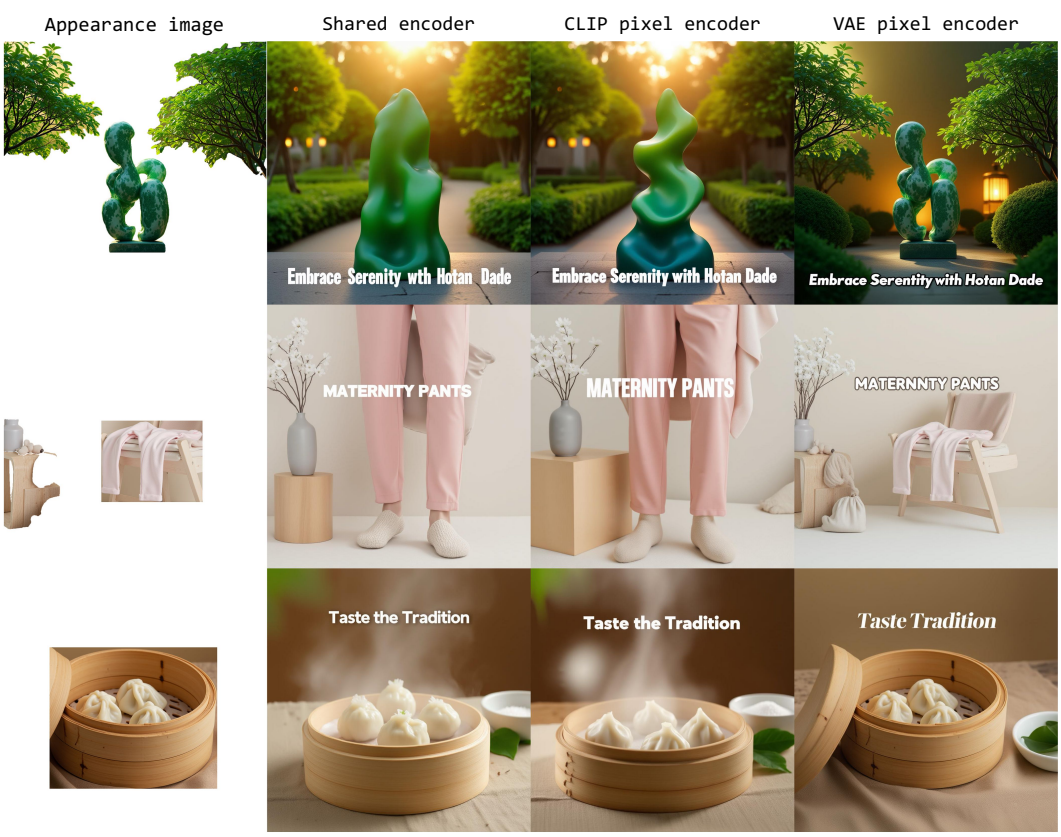


Figure 10: **Qualitative comparison of pixel encoding methods.** Generated outputs illustrate differences in preserving reference appearance.

per-token hidden states after the vision tower's layer normalization, rather than its pooled contrastive projection. DeepEncoder V2 uses a 1024^2 global view and local 768^2 crops, producing 256 to 1,120 tokens depending on the number of crops. The adapters are first aligned on captions for 1,000 steps, followed by 85,000 steps on internally assembled image–caption data at 256^2 resolution. Each arm then follows the same CreatiDesign curriculum with 2,000 steps at 256^2, 5,000 steps at 512^2, and 3,000 steps at 1024^2. We evaluate the resulting checkpoints on the same 152 square designs with 50 sampling steps and CFG 7. This comparison changes the full pretrained reader, including its architecture and token budget, while holding the downstream training and evaluation recipe fixed.

O Pixel Encoder Qualitative Comparison

Figure 10 compares generated subjects under different pixel encoding methods, complementing the subject-fidelity measurements in Table 3. In the shared-encoder baseline, DeepEncoder reads both semantic conditions and appearance references. The CLIP and VAE configurations instead use a dedicated pixel encoder alongside the semantic encoder.

P Additional Generated Designs

Figure 11 provides additional examples generated with All-in-Image, complementing the controlled comparisons in Figure 4.



Figure 11: **Additional graphic designs generated with All-in-Image.** Selected outputs include product advertisements, interiors, food, and lifestyle scenes. These examples illustrate the application of the proposed conditioning method to different design inputs.



Figure 12: **Generation with and without a global prompt.** Each pair retains the same regional instructions, bounding boxes, and appearance image, while the global prompt and its full-image box are included or removed. Sampling settings are fixed across the pair.

Q Flexible Semantic and Appearance Conditioning

We examine how global guidance and reference placement affect generation. All examples use guidance scale 4, 50 denoising steps, and seed 42.

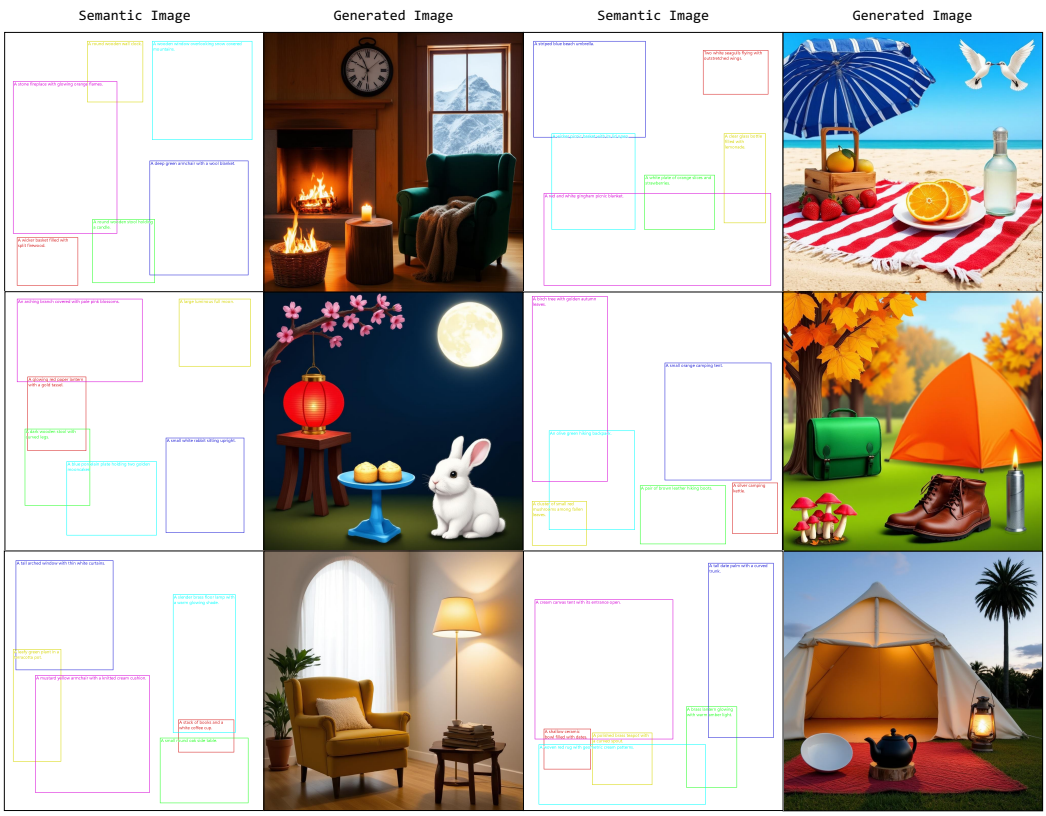


Figure 13: **Generation from regional prompts alone.** Each semantic image contains six regional descriptions and their boxes, with no global prompt. The appearance image is uniformly white in every case, with no subject or glyph references. Selected outputs illustrate scene composition from spatially organized local instructions.

Generation without a global prompt. For selected CreatiDesign examples, we remove the global prompt and its full-image box while retaining regional instructions, their boxes, and the original appearance image. Figure 12 compares this setting with the standard inputs. The juice-bottle and boots examples retain their principal subjects and broad arrangement, while typography and scene details change. These examples show that regional instructions and appearance references can guide generation without an additional global description.

Generation from regional prompts alone. We further construct scenes specified by six regional prompts and their bounding boxes. The semantic image contains no global prompt, and the appearance image is uniformly white, with no subject or glyph references. These examples use the same checkpoint. Figure 13 shows selected semantic images and outputs. In the moon festival rabbit example, the model combines the specified branch, lantern, stool, mooncakes, and rabbit into an interior. The example illustrates scene composition from regional instructions without a global description or appearance reference.

Concept guidance and appearance preservation. We also compare placing the same subject references in the appearance image or directly in the semantic image. In the standard setting, the pixel encoder reads the references from the appearance image. In the alternative zero-shot setting, we paste the subject references into the semantic image at their target locations, alongside the original written instructions, so that they are read by the semantic image encoder. The appearance image either retains only the glyph references or is replaced by a uniformly white image. Figure 14 illustrates the resulting difference. The standard setting preserves the distinctive reference shapes

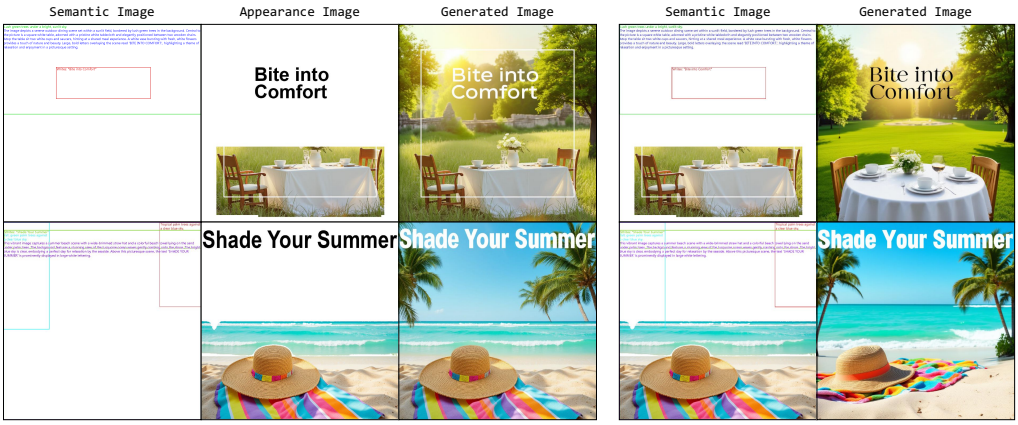


Figure 14: **Zero-shot concept guidance and appearance preservation.** Subject references are encoded either from the appearance image by the pixel encoder or from the semantic image by the semantic encoder. For the latter, the appearance image contains only glyph references or is uniformly white, as indicated. The merged semantic input retains the broad object concepts while allowing changes in shape and detail. For example, the rectangular dining-table reference produces a round table when placed in the semantic image.

more closely, whereas the merged semantic input retains the broad subject concepts while allowing changes in shape and detail. For example, the rectangular dining-table reference produces a round table when placed in the semantic image. These examples distinguish reference-appearance preservation from using reference images as broader visual cues.

R Visual Conditioning Interface

Figure 15 illustrates a visual interface for composing the conditions of All-in-Image. Users arrange regional descriptions and their boxes, text-rendering instructions, subject and glyph references, and freehand sketches on a shared canvas. The interface derives and previews the semantic and appearance images from this composition. The global prompt is edited separately but rendered into the semantic image, without an additional text-conditioning pathway. In the illustrated example, regional instructions describe the daisies, blueberries, and moss, while a butterfly sketch provides a visual cue. The perfume bottle and headline glyphs appear in the appearance image. The resulting advertisement is displayed alongside the input composition, showing how these different conditions can be combined in one workflow.

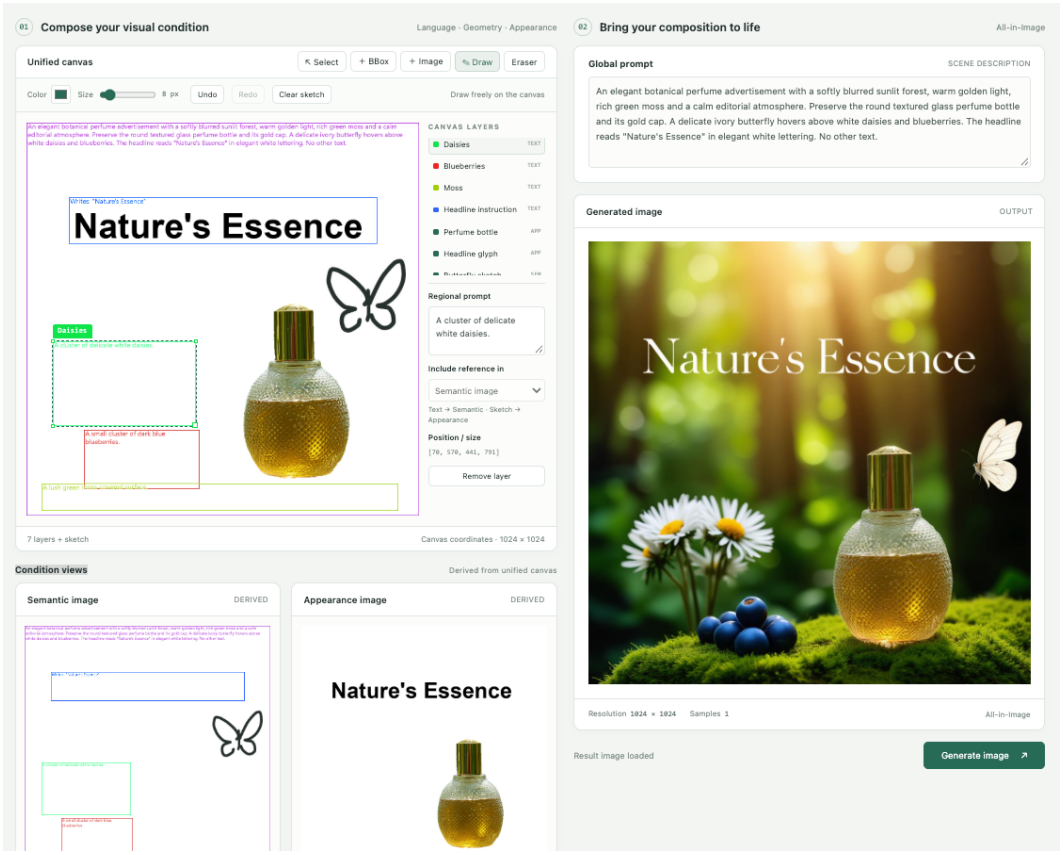


Figure 15: **Visual conditioning interface for All-in-Image.** Left: a shared editing canvas and previews of the derived semantic and appearance images. Right: the global-prompt editor and the generated design. The example combines regional instructions, a hand-drawn butterfly, and subject and glyph references in a perfume advertisement.