DUALFLOW: DUAL DIFFUSION BRANCHES FOR CROSS-MODAL INFORMATION FLOW IN GRAPHIC DESIGN TEMPLATE GENERATION

Anonymous authors
Paper under double-blind review

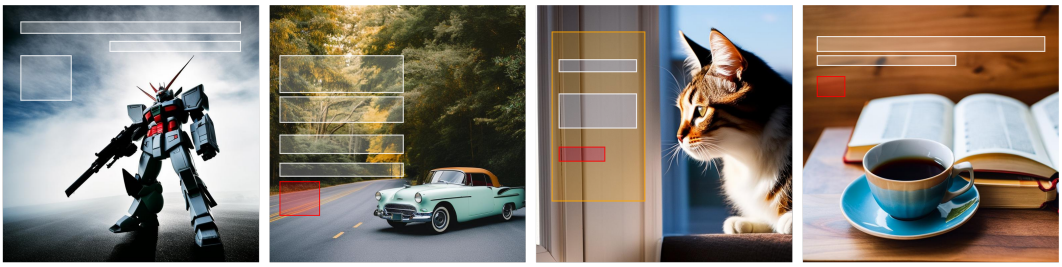


Figure 1: Generated design templates by our model. Each template has a background image and a layout of elements: text (white), underlay (yellow), and button (red). Input text description is shown below each template.

ABSTRACT

In this paper, our goal is to automate design template creation that generates a background image and a layout of foreground elements over the background to form a harmonious composition from an input text. Prior works on design template generation opt to generate the background and layout sequentially using two separate models and model the dependency between them by simply conditioning one model on the final output of the other. Hence, these methods fall short of capturing intricate interaction between the background and layout. To overcome this limitation, we propose a diffusion model, *DualFlow*, which jointly generates background images and layouts in a single generative process. The novel design of our joint model's denoising network connects the backbones of pre-trained image and layout diffusion models with a carefully designed, learnable communication module. At training time, the image and layout backbones are frozen to maintain the pre-trained priors, while the communication module is trained from scratch to focus on learning subtle image-layout interaction to generate more harmonious compositions. Furthermore, we introduce two metrics, *TemplateFID* and *TemplateCLIP*, to assess the quality of generated design templates holistically. Our experiments show that, compared with prior approaches, our model can achieve significantly better results, and produce outputs that are closer to real samples. We also demonstrate the flexibility of our model in enforcing additional design principles at inference without retraining.

1 INTRODUCTION

When creating graphic designs, designers usually choose and start from pre-existing design templates, instead of designing from scratch, and iteratively refine the templates towards the final full designs. Design templates are of great value to designers in practice, which can provide inspirational ideas and help speed up the design process. However, creating good design templates is non-trivial, which often takes a lot of manual efforts and necessitates considerable design expertise.

In this paper, we aim to automate the process of creating graphic design templates from input textual descriptions. Design templates usually lie midway between high-level requirements and full designs:

they are more concrete than high-level intentions with basic appearance and structure to serve as good starting points, while being less complete than full designs with only a partial set of element attributes to leave room for creativity. Following the problem setup of Weng et al. (2024), we assume that a design template is composed of a background image and a layout (spatial arrangement) of foreground elements over the background.

A recent work, Desigen Weng et al. (2024), presents a solution to the design template generation problem. Desigen takes a two-stage approach, where a background image is first generated, followed by generating a layout of elements conditioned on the background image. It also implements an iterative refinement of the generated image and layout by repeating the two-stage procedure, where the output of one stage is fed back into the other stage. Despite encouraging results, this method suffers from several shortcomings. First, it models image and layout *separately*, and the communication between the image and layout models is limited since only the end output of one model is used to influence the other model. Consequently, It has limited ability to capture complex interaction between image and layout, which is crucial for harmonious compositions. Second, it explicitly introduces an inductive bias towards occlusion avoidance, to reserve empty space in generated images for foreground object placement. This will cause the distribution of generated samples drift significantly from training sample distribution, washing out distinctive patterns of training samples from the model outputs.

To mitigate this issues, we propose *DualFlow*, a latent diffusion model for design template generation, which enables *joint* generation of images and layouts in a *single* framework by learning nuanced image-layout interaction. Specifically, we first pretrain a latent diffusion model for unconditional layout generation as *layout prior*, and leverage a pretrained text-to-image latent diffusion model Rombach et al. (2022) as *image prior*. Each of the two priors alone can generate high-quality samples in its own domain (image or layout). We then construct the denoising network of DualFlow by combining the pretrained backbones of the two priors and connect them with a communication module dedicated to model image-layout interaction. When training DualFlow on design template data, we hold the pretrained weights of the two backbones fixed, and only optimize the communication module. This allows us to focus learning on capturing intricate image-layout interaction patterns, as the prior knowledge of generating realistic images and layouts are already available in the two backbones. With the learned communication module, our DualFlow can generate a holistic design template with a single denoising process, through which the image and layout backbones frequently communicate with each other to denoise noisy image and layout latents jointly. Therefore, our model is able to produce more harmonious compositions of image and layout compared to Desigen. Besides, while our model refrains from using any design-related inductive bias, thus better fitting the training data distribution. It offers great flexibility in enforcing some design principles in generated design templates by utilizing a simple guidance technique *at test time*.

For assessing the performance of design template generation models, the prior study Weng et al. (2024) measures image quality and layout quality separately, without evaluating design templates holistically. Hence, we introduce two evaluation metrics, *TemplateFID* and *TemplateCLIP*, which respectively assess the visual quality and text adherence of generated templates. The proposed metrics complement existing image and layout metrics, enabling more comprehensive evaluation of design template generation models.

To validate the effectiveness of our DualFlow model, we conduct experiments on the Web-design dataset Weng et al. (2024), comparing DualFlow with state-of-the-art approaches. Our model exhibits dramatically improved performance compared to the prior approaches, generating high-quality design templates (see Figure 1). We also demonstrate that our model can optionally impose an occlusion avoidance rule similarly to Desigen, effectively reducing the occlusion of salient background regions by foreground elements.

The main contributions of this paper are threefold: we propose a diffusion-based design template generation model that synthesizes images and layouts simultaneously in a single process by learning the interaction between pretrained, fixed image and layout priors, in sharp contrast to the two-stage strategies used in prior work; we introduce a simple yet effective training-free sampling strategy that enables prioritization of specific design rules during generation without retraining; and we design two quantitative metrics to holistically assess the visual quality and text alignment of design templates.

2 Related Work

2.1 Graphic Layout Generation

Layout generation, which involves creating spatial arrangement of elements on a canvas, is fundamental task in graphic design and has been extensively studies in recent years. Early works have approached layout generation using various generative modeling frameworks including Generative Adversarial Networks (GANs) Li et al. (2019); Zheng et al. (2019); Kikuchi et al. (2021) and Variational Autoencoders (VAEs) Jyothi et al. (2019); Arroyo et al. (2021), Transformers Gupta et al. (2021); Horita et al. (2024); Jiang et al. (2023), diffusion models Inoue et al. (2023); Zhang et al. (2023); Cheng et al. (2023) and flow-based models Guerreiro et al. (2024). Previous works also show the effectiveness of large language models (LLMs) in solving the layout generation tasks based on the layout-related knowledge that LLMs have acquired during pretraining.

Besides unconditional generation, constrained layout generation has been investigated, which imposes various user constraints on element attributes and relationships to control generated layouts Lee et al. (2020); Kikuchi et al. (2021); Jiang et al. (2023). Several recent methods investigate graphic design composition, which uses a set of multimodal elements (images and texts) as conditioning information, and compose them into a cohesive design Cheng et al. (2024); Shabani et al. (2024); Zhang et al. (2025); Lin et al. (2025). Our work is related to a branch of methods on content-aware layout generation, which generates layouts conditioned on a background image Zhou et al. (2022); Cao et al. (2022); Horita et al. (2024). However, instead of tackling background-conditioned layout generation solely, we aim to generate both layout and background image, in order to constitute a design template.

2.2 Graphic Design Generation

Compared with layout generation, building generative models for complete graphic designs remains a less explored problem. CanvasVAE Yamaguchi (2021) trains a VAE to generate vector graphic designs represented as sets of canvas and element attributes. GOL Yang & Cao (2025) shows that learned element order can improve the performance of autoregressive and diffusion generators that predict a sequence of element attribute tokens.

Recently, several methods have been proposed to automate graphic design generation from text prompts specifying design intention. One class of methods directly fine-tune pretrained text-to-image diffusion models for text-to-design generation, producing highly aesthetic and coherent design images. Another class of methods try to generate layered graphic designs, composed of multiple image and text layers, using either a cascade of task-specific models Jia et al. (2023); Inoue et al. (2024) or specialized multi-layer image generation models Pu et al. (2025).

The design template generation problem that we focus on can be viewed as a specialization of design generation that considers a subset of element attributes, including background image and element layout. Desigen Weng et al. (2024) is the initial attempt for design template generation. it introduces an inductive bias for minimizing occlusion between foreground layout elements and salient background objects, which may cause a gap between the learned distribution and the training data distribution. In contrast, our model generates a background image and a layout jointly in a single diffusion model, which focuses on learning image-layout interaction to achieve more coherent image-layout composition. In addition, our model imposes no design-specific inductive bias, thereby better maintaining the distinctive features of training samples in generated design templates.

3 Method

Given a text description $\mathcal{P}$, our goal is to generate a design template X^D. Each design template consists of a background image X^I and a layout X^L, where the layout comprises a set of elements defined by their categories and bounding box coordinates. To this end, we aim to learn a joint distribution $p(X^I, X^L|\mathcal{P})$ of background images X^I and layouts X^L conditioned on $\mathcal{P}$, from which we can sample X^I and X^L, and compose them into a coherent design template. The key idea of our method is to 1) train two expressive diffusion priors on image and layout domains (to ensure the high-quality generation of background images and layouts), and 2) learn the interaction between the

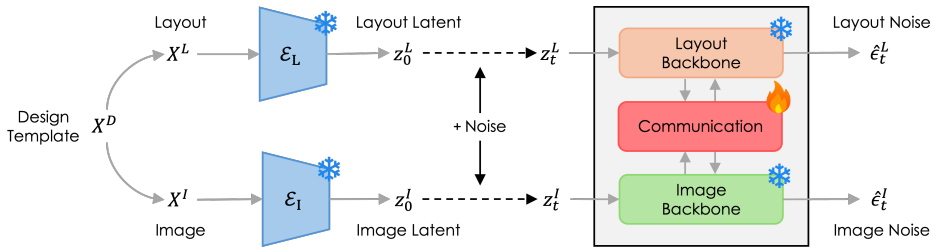


Figure 2: Overview of our joint model.

two priors in a single joint diffusion model (to ensure a harmonious composition). This leads to an unified model *DualFlow* for joint generation of background images and layouts from text inputs.

3.1 Domain-specific Priors

We aim to build a conditional image prior $p(X^I|\mathcal{P})$ that can generate images from text prompts, and a layout prior $p(X^L)$ for generating high-quality layouts.

Image Prior. For $p(X^I|\mathcal{P})$, we employ Stable Diffusion (*SD*) Rombach et al. (2022), a large-scale latent diffusion model (LDM) for text-to-image generation. SD trains a variational autoencoder (VAE) with an image encoder $\mathcal{E}_\text{I}$ and an image decoder $\mathcal{D}_\text{I}$. The encoder $\mathcal{E}_\text{I}$ maps a RGB image $X^I \in \mathbb{R}^{H\times W\times 3}$ to a spatial latent representation $z_0^I = \mathcal{E}_\text{I}(X^I) \in \mathbb{R}^{h\times w\times d^I}$ (a $h \times w$ grid of embeddings, each of dimensionality d^I). Then, a diffusion model is learned over latents instead of pixels. For image generation, a latent $\tilde{z}_0^I$ is sampled from the diffusion model and then decoded by the decoder $\mathcal{D}_\text{I}$ into an image $\tilde{X}^I = \mathcal{D}_\text{I}(\tilde{z}_0^I)$. To adapt the pretrained SD to the task of generating background images in graphic designs, we fine-tune it on a dataset of background images associated with text descriptions, to obtain the adapted denoising network ϵ_{θ^*}, which we refer to *image backbone* in our joint model introduced later.

Layout Prior. For $p(X^L)$, we train another LDM from scratch on a layout dataset. Following the prior work on layout generation Gupta et al. (2021); Jiang et al. (2023); Inoue et al. (2023), we define a layout element as a bounding box with 5 attributes including category, left coordinate, top coordinate, width and height. After discretizing the continuous attributes into bins using the k-means algorithm, an element is formatted as a sequence of 5 attribute tokens, and a layout is the concatenation of the element sequences. We first train a VAE to project layouts into a latent space. The Transformer-based VAE encoder $\mathcal{E}_\text{L}$ encodes a layout X^L into a latent representation $z_0^L = \mathcal{E}_\text{L}(X^L) \in \mathbb{R}^{N\times d^L}$, where N is the number of elements in the layout and d^L is the embedding dimensionality. We then train a diffusion transformer Peebles & Xie (2023) over the latent space. A layout can be generated by sampling $\tilde{z}_0^L$ from DiT, followed by feeding it into the Transformer-based VAE decoder $\mathcal{D}_\text{L}$: $\tilde{X}^L = \mathcal{D}_\text{L}(\tilde{z}_0^L)$. We refer to the trained Transformer-based denoising network ϵ_{ϕ^*} of DiT as *layout backbone* in our subsequent joint model.

3.2 Joint Model

Given the pretrained two priors that operate independently, we merge them into a single diffusion model to simultaneously generate background images and layouts. As illustrated in Figure 2, the core design of the joint model's denoising network is to put the *pretrained* and *frozen* image and layout backbones together to jointly denoise image and layout noisy latents, while connecting the backbones using a *learnable* communication module to capture image-layout interaction. Such design has two primary benefits: first, by locking the weights of the backbones, we can retain the pretrained knowledge in them to effectively leverage their original capabilities to ensure the quality of generated images and layouts; second, as the prior knowledge of how to generate images and layouts independently is already available in the two pretrained backbones, our joint model, when trained on design template datasets, can focus on learning intricate and subtle interaction between image and layout, thus improving composition harmony. Furthermore, the two backbones can communicate over a large number of denoising steps *during* the generation process of images and layouts. This is different from the sequential iterative refinement (e.g., in Desigen), where the

image (layout) is generated *after* the generation of the layout (image). Consequently, our joint model can better model image-layout interaction, compared to the previous work.

To learn our model, we train a joint denoising network:

$$[\hat{\epsilon}_t^I, \hat{\epsilon}_t^L] = \epsilon_{\psi,\theta^*,\phi^*}(z_t^I, z_t^L, t, \mathcal{P}), \tag{1}$$

where $\hat{\epsilon}_t^I, \hat{\epsilon}_t^L$ are predicted noises. The training is performed by optimizing the following objective:

$$\mathcal{L} = \mathbb{E}_{z_0^I, z_0^L, t, \epsilon_t^I, \epsilon_t^L}\Big[\lambda_I \|\epsilon_t^I - \hat{\epsilon}_t^I\|_2^2 + \lambda_L \|\epsilon_t^L - \hat{\epsilon}_t^L\|_2^2\Big], \tag{2}$$

where $t \sim [1, T]$ (T is the number of diffusion timesteps), $\epsilon_t^I \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\epsilon_t^L \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Our implementation sets $\lambda_L = 3$ and $\lambda_I = 1$. ψ contains the communication module's trainable weights.

3.3 Communication Module

The communication module is designed to facilitate bidirectional information exchange between the image and layout backbones, while preserving the their pretrained knowledge as much as possible. Let $h^I \in \mathbb{R}^{M \times d}$ be the flattened intermediate representations of the image backbone (U-Net), and $h^L \in \mathbb{R}^{N \times d}$ be the intermediate representation after the 20th DiT block of the layout backbone. The communication module augments h^I and h^L as:

$$[\bar{h}^I, \bar{h}^L] = \mathrm{COMM}(h^I, h^L). \tag{3}$$

The augmented representations $\bar{h}^I, \bar{h}^L$ are then fed into the subsequent blocks in the image and layout backbone. More specifically, to obtain $\bar{h}^I$, the operations can be written as:

$$\bar{h}^I = h^I + \eta \cdot \mathrm{CrossAttn}(h^I, h^L). \tag{4}$$

The cross-attention can model image-layout interaction, propagating layout information from h^L to h^I. η is set to 1 during training and can be varied during inference to control when the communication is enabled over the denoising process. Similarly, $\bar{h}^L$ is obtained by:

$$\bar{h}^L = h^L + \eta \cdot \mathrm{CrossAttn}(h^L, h^I). \tag{5}$$

When training the model, we set $\eta = 1$ in Equation 4 and 5, enabling the communication module for all diffusion timesteps so that this module can be sufficiently updated. However, our early experiments show that using $\eta = 1$ throughout the entire denoising process at test time leads to unsatisfactory layout generation quality. To mitigate this issue, inspired by GLIGENLi et al. (2023), we dynamically adjust the value of η during the sampling process to improve visual quality. The denoising process of diffusion models generates coarse-grained features (e.g., global structures and colors) at early steps, and then generates perceptually important contents and fine-grained details at late steps. Therefore, we opt to turn on the communication module ($\eta = 1$) at the first 30% of the denoising process, where the image backbone can use *external* knowledge from the layout backbone to generate rough spatial arrangement of objects in the image that harmonizes with the layout, and vice versa for the layout backbone. In the remaining denoising steps, we disable the communication module ($\eta = 0$), allowing the two backbones to rely upon their *internal* knowledge to generate high-quality images and layouts. A detailed ablation of different η values during inference is provided in Appendix A.5.

3.4 Guidance based on Design Principles

To offer flexibility in enforcing design principles on generated samples, we propose a simple guidance technique to guide the layout generation process by adjusting the noise prediction $\hat{\epsilon}_t^L$ of the layout backbone:

$$\hat{\epsilon}_t^L \leftarrow \hat{\epsilon}_t^L - s\sqrt{1-\bar{\alpha}_t}\,\nabla_{z_t^L} \sum_{k=1}^{K} \mathcal{L}\big(\hat{\epsilon}_t^I, \hat{\epsilon}_t^L\big), \tag{6}$$

where s is the guidance scale, and $\bar{\alpha}_t$ is the multiplication of the forward process noise variances from 1 to t. $\{\mathcal{L}_k\}$ are a group of differentiable loss functions the penalize the violation of some design principles. $[\hat{\epsilon}_t^I, \hat{\epsilon}_t^L] = \epsilon_{\psi,\theta^*,\phi^*}(z_t^I, z_t^L, t, \mathcal{P})$, where ψ denotes the trained weights by minimizing Equation 2. Note that any design principle can be imposed here as long as their corresponding differentiable loss function can be defined.

As an example, we show how to define a loss function for an occlusion avoidance rule that prevents foreground elements from occluding salient background regions. Specifically, given the predicted image noise $\hat{\epsilon}_t^I$, we compute a clean image latent $\hat{z}_0^I$ analytically using the forward process marginal distribution $q(z_t^I|z_0^I)$, decode it into an image $\mathcal{D}_{\text{I}}(\hat{z}_0^I)$, and compute a saliency map $S = f_{\text{sal}}(\mathcal{D}_{\text{I}}(\hat{z}_0^I))$ using an off-the-shelf saliency detector f_{sal}. Then, we compute a clean layout latent $\hat{z}_0^L$ from the predicted layout noise $\hat{\epsilon}_t^L$ with $q(z_t^L|z_0^L)$, and decode it into a layout $\mathcal{D}_{\text{L}}(\hat{z}_0^L)$. The occlusion avoidance loss $\mathcal{L}_{\text{occ}}$ is computed as the average saliency value within the decoded layout elements on S. Let $\{\mathbf{b}_i\}_{i=1}^B$ be a set of decoded element bounding boxes. Formally, $\mathcal{L}_{\text{occ}}$ is written as:

$$\mathcal{L}_{\text{occ}} = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{A_i}\sum_{p} M_{\mathbf{b}_i}(p)\, S(p), \tag{7}$$

where $M_{\mathbf{b}_i}$ is a soft mask for $\mathbf{b}_i$ with higher values at locations within $\mathbf{b}_i$, A_i is the area of $\mathbf{b}_i$, and p indexes spatial positions. See more details in the supplementary material.

4 Experiments

4.1 Dataset and Implementation Details

We conduct our experiments on the Web-design dataset Weng et al. (2024), which consists of 50K web banner designs collected from real-world online shopping platforms. We use 41,270 samples (85%) for training, 2,427 (5%) for validation, and 4,856 (10%) for testing. For pretraining the layout prior, we train its VAE (embedding dimensionality 32) using β-VAE with $\beta = 5 \times 10^{-4}$. The layout backbone is trained for 1000 epochs with a batch size of 4096. For the image prior, we fine-tune the image backbone for 100 epochs with a learning rate of 1×10^{-5} on the background images from the Web-design. More detailed settings are provided in Appendix A.7 and A.8 .

4.2 Compared Methods

We compare our method with a stat-of-the-art design template generation model, *Desigen* Weng et al. (2024), which is already trained on the Web-design dataset. Desigen proposes an iterative strategy to refine the generated image and layout. We run this iterative refinement for 3 iterations in the comparison. We also compare with the sequential text-to-design model *OpenCOLE* Inoue et al. (2024), and a pipeline that combines vanilla Stable Diffusion (SD) Rombach et al. (2022) for background generation with GPT-4o (denoted as GPT-4V) for layout inference, serving as a vision–language baseline.

4.3 Evaluation Metrics

Domain-specific Metrics. Following the evaluation protocol of Weng et al. (2024), we evaluate the quality of background image and layout seperately using the following metrics. For background image evaluation, we use: *Saliency Ratio* Weng et al. (2024) that measures the proportion of salient regions in an image; *FID* Heusel et al. (2017) that measures visual quality and is computed against the training split; *CLIP Score* Radford et al. (2021) that measures text-image alignment. For layout evaluation, we use: *Alignment* Li et al. (2019) that measures alignment between layout elements; *Overlap* Li et al. (2019) that measures the amount of overlap between layout elements; *Occlusion* Cao et al. (2022) that measures the occlusion of background salient regions by layout elements. As additional layout metrics for comprehensive evaluation, We also include: *LayoutFID* Kikuchi et al. (2021) that measures overall layout quality and is computed against the training split; *Readability* Horita et al. (2024) that measures text readability.

Holistic Metrics. We further propose two metrics, *TemplateFID*, *TemplateCLIP*, to evaluate the quality of holistic design templates, which consider background image and layout jointly. TemplateFID evaluates how realistic generated templates; TemplateCLIP tells how well generated templates

adhere to the text inputs. To compute the metrics, we first train an template autoencoder (AE) on the Web-design dataset to extract joint image-layout (or template) embeddings. Given a background image and a layout, the encoder maps them to a single template embedding. For this, we first embed the background image and layout using the pretrained image encoder $\mathcal{E}_{\mathrm{I}}$ and layout encoder $\mathcal{E}_{\mathrm{L}}$ (of the image and layout priors), combine the image and layout embeddings, and process them via a series of Transformer encoder blocks. The decoder reconstructs the image embeddings and the layout from the template embedding, using another series of Transformer encoder blocks, followed by linear projection layers. To further validate the effectiveness of TemplateFID, we conduct a user study on template embedding similarity (Appendix A.4) and find that Participants preferred the retrieved templates over random ones 73.8% of the time, suggesting that the learned embedding space indeed captures meaningful structural characteristics of design templates.

To compute TemplateFID, we calculate Fréchet Inception Distance (FID) Heusel et al. (2017) between generated and real samples based on the template embeddings. To compute TemplateCLIP, we fine-tune the template AE encoder and the pretrained SD text encoder with the contrastive learning objective of the CLIP Radford et al. (2021) on the Web-design dataset.

Method	Image			Layout					Template		Inference Time (s/template)
	FID↓	CLIP↑	Saliency Ratio	LayoutFID↓	Align	Overlap	Occlusion	Readability	TemplateFID↓	TemplateCLIP↑	
Desigen (0)	31.52	29.20	20.65%	0.56	0.35	14.41	13.47%	11.48%	118.96	3.19	7.47
Desigen (1)	30.84	28.87	19.29%	0.57	0.38	15.63	13.24%	11.56%	108.56	3.18	14.85
Desigen (2)	31.23	28.46	18.36%	0.51	0.37	15.52	12.70%	11.23%	114.10	3.19	22.42
Desigen (3)	30.79	29.04	17.67%	0.48	0.37	15.23	11.45%	10.97%	116.83	3.21	29.89
GPT-4V	22.90	**31.297**	28.77%	9.457	0.01	99.52	30.49%	12.52%	237.68	2.98	10.86
Ours	**19.57**	29.79	**14.56%**	**0.15**	**0.31**	**11.98**	**19.30%**	11.41%	**86.63**	**3.25**	**5.01**
Ours + OAG	**19.57**	29.79	**14.56%**	0.21	0.32	13.26	12.66%	**10.35%**	92.81	3.23	8.69
Real Data	–	27.5	14.17%	–	0.31	9.34	21.14%	8.53%	–	3.39	–

Table 1: Quantitative comparison with Desigen and GPT-4V on the Web-design dataset. "Desigen (n)" denotes applying iterative refinement n times for Desigen. For all metrics except CLIP and FID-related ones, values closer to the ones calculated from real data at the bottom row indicate better performance. The best and second best results are in bold and underlined, respectively.

Method	Image			Layout					Design		Inference Time (s/template)
	FID↓	CLIP↑	Saliency Ratio	LayoutFID↓	Align	Overlap	Occlusion	Readability	TemplateFID↓	TemplateCLIP↑	
OpenCOLE	23.79	30.49	23.24%	1.63	4.20	**11.44**	30.48%	11.74%	211.42	3.12	25.8
Ours (Prompt Aug)	**18.97**	**30.94**	**15.32%**	**0.16**	**0.32**	12.06	**19.74%**	**11.63%**	**85.35**	**3.28**	**10.14**
Real Data	–	27.5	14.17%	–	0.31	9.34	21.14%	8.53%	–	3.39	–

Table 2: Quantitative comparison with OpenCOLE on the Web-design dataset. The text prompts input to our model are augmented by GPT. For all metrics except CLIP and FID-related ones, the values closer to the ones calculated from real data at the bottom row indicate better performance. The best and second best results are in bold and underlined, respectively.

4.4 Quantitative Results

Comparison to Desigen and GPT4V. Table 1 presents the quantitative comparison with Desigen. We report metrics computed on the test set of the Web-design dataset as a reference, shown as *Real Data* at the bottom of the Table. 1. For image generation, our method outperforms Desigen across all the three metrics, with substantial improvements in FID and saliency ratio. This indicates that our model can generate high-fidelity background images with sufficient empty space for foreground element placement. It should be noted that the CLIP score of real data is lower than other methods, as real background images leave more empty space for foreground elements (lower saliency ratio), which weakens text–image alignment. By contrast, GPT-4V achieves the highest CLIP score due to minimal empty space in SD-generated images, but its predicted layouts deviate sharply from real data and suffer from extremely high overlap. For layout generation, our model achieves the best results on 4 out of the 5 the metrics, and is comparable to Desigen for the readability metric, demonstrating its superior layout generation capabilities. Also, our model is able to generate more harmonious design templates that more faithfully conform to the input texts, as evidenced by its better results in TemplateFID and TemplateCLIP.

As Desigen explicitly enforces occlusion avoidance constraints in its model design, its saliency ratio and occlusion scores decline over refinement iterations, meaning that the salient portion of the generated background image is gradually reduced, and layout elements and salient background regions are

moved further apart from each other. While such iterative refinement boosts performance in some metrics (e.g., LayoutFID and readability), a side effect is exposed: the gaps between Desigen's generated samples and real samples, in terms of several aspects (measured by metrics, e.g., alignment, overlap and occlusion), become increasingly larger. The growing gaps imply that the refined results lose some unique design characteristics in training examples, which is undesirable. For example, designers may occasionally make foreground elements partially occlude salient background objects, e.g., for creative or artistic purposes. Excessive occlusion reduction can lead to a lower occlusion score, but may generate unnatural outputs that don't resemble what human designers create. Due to the existence of the aforementioned gaps, Desigen's TemplateFID improves only slightly with the iterative refinement, still significantly lagging behind that of our model. In addition, Desigen's iterative refinement runs the denoising process of the SD multiple times, which incurs non-trivial computational overhead. In contrast, our model only requires a single pass of the denoising process, which gives it a significant advantage in inference time.

Occlusion Avoidance Guidance. Our model introduces no design-specific inductive bias, but allows for enforcing design principles in the generation process through inference-time guidance. We have implemented occlusion avoidance guidance (OAG) that prevents important background regions from being blocked by foreground elements similarly to Design, and report its results in Table 1. Applying OAG leads to a considerable reduction of the occluison metric, which showcases its effectiveness(see Appendix A.3 for detailed quantitative and qualitative analyses).

Comparison to OpenCOLE. OpenCOLE consists of three modules: 1) a design plan generation module (a pretrained large language model) to convert a brief design intention into a detailed design plan; 2) an image generation module (a fine-tuned SD model) that generates an image (i.e, a background image in our problem setting) conditioned on the concatenation of object and background captions from the design plan; 3) a typography generation module (a fine-tuned large multimodal model) that generates text attributes (including the bounding box parameters of texts) based on the design plan and the generated image. For design template generation, OpenCOLE can be understood as a two-stage approach (image generation followed by text layout generation) similar to Desigen, with an additional component (the design plan generation module) to augment the input text prompt. For a fair comparison, we modify the image generation module to use the same SD model as our model, and change the typography generation module to output the layout of elements. The above two modules are fine-tuned on the Web-design dataset. Furthermore, we leverage GPT to turn short text inputs into long, detailed prompts before feeding them into our model. Table 2 report the quantitative results of OpenCOLE and our model. It can be seen that our model outperforms OpenCOLE on 9 out of the 10 metrics. Additionally, we conduct a GPT-4V evaluation (Appendix A.2), leveraging a vision–language model to assess the generated templates. Table 4 show that our method produces design templates that are more coherent, semantically accurate, and visually consistent.

4.5 Qualitative Results

Figure 3 show the visual comparison of design templates generated by different methods. Our model can generate high-quality diverse layouts and visually pleasing background images. More importantly, due to its strong ability to capture intricate interaction between layout and background image, our model can produce harmonious layout-image compositions. In contrast, the results of OpenCOLE , Desigen and GPT-4V suffer from various issues. Specifically, in the results of OpenCOLE, foreground elements are significantly misaligned and salient background areas are often occluded by foreground elements. The layouts of Desigen sometimes exhibit undesirable overlap between elements (column 2, 5) and repetitive patterns (e.g., la) the background images generated by Desigen appears less visually aesthetic. Moreover, GPT-4V tends to predict layouts that spread across the entire background image, leading to excessive occlusion and unrealistic template structures.

Method	Image			Layout					Template	
	FID↓	CLIP↑	Saliency Ratio	LayoutFID↓	Align	Overlap	Occlusion	Readability	TemplateFID↓	TemplateCLIP↑
wo/ Comm.	**19.32**	**30.37**	21.37%	0.16	0.38	12.87	30.74%	12.23%	99.14	3.23
w/ Comm.	19.57	29.79	**14.56%**	**0.15**	**0.31**	**11.98**	**19.30%**	**11.41%**	**86.63**	**3.25**
Real Data	–	27.5	14.17%	–	0.31	9.34	21.14%	8.53%	–	3.39

Table 3: Ablation study on the communication module. For all metrics except CLIP and FID-related ones, the values closer to the ones calculated from real data at the bottom row indicate better performance. The best results are in bold.

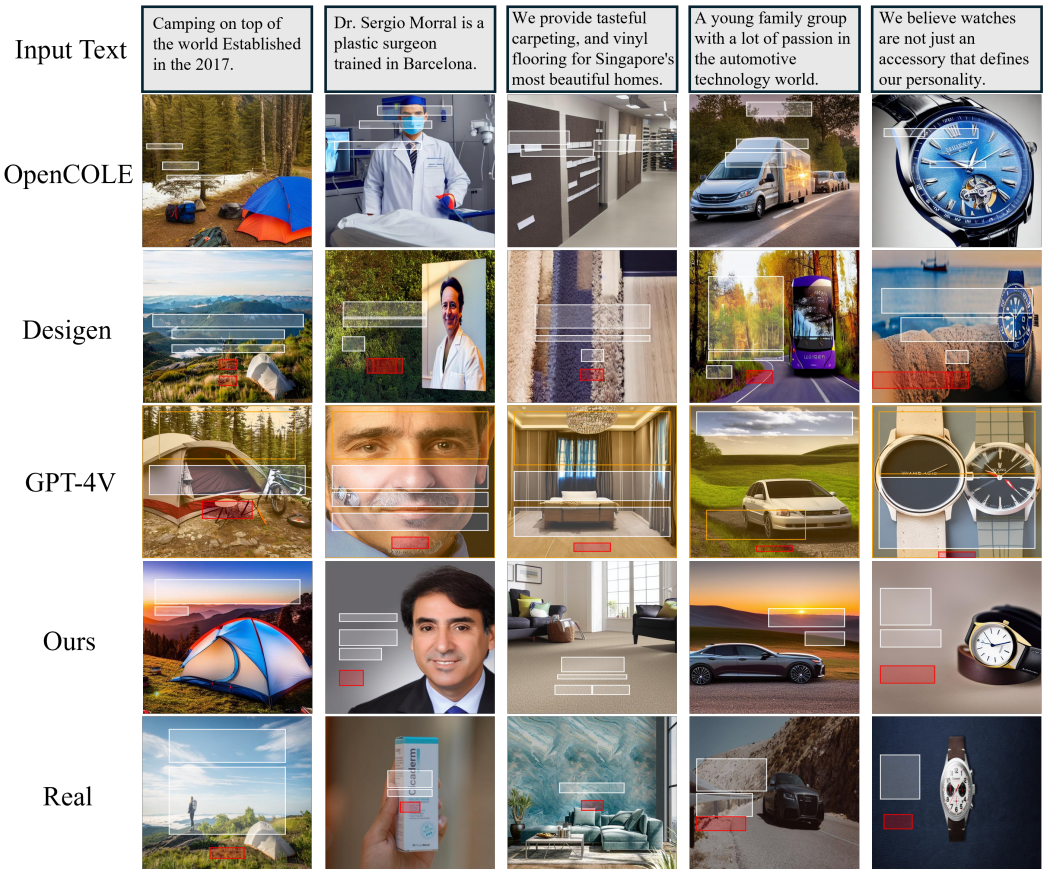


Figure 3: Qualitative comparison of different methods. White, orange and red boxes denote text, underlay and button, respectively. Input texts are shown at the top of each column, and the last row displays the ground truth design templates.

4.6 Importance of the Communication Module

One key ingredient of our model is the communication module that enables the image and layout backbones interact with each other during the generation process. To test the effect of this component, we consider a variant of our model where the communication module is disabled and thus the two backbones denoise *independently*. As shown in Table 3, using the communication module dramatically reduces the saliency ratio, occlusion and readability, and contributes to a significant improvement of TemplateFID. This highlights the importance of the communication module for improving the composition harmony of background image and layout. Additionally, to gain a more intuitive understanding of what the communication module learns, we provide visualizations of its cross-attention maps in different stages and qualitative comparisons in Appendix A.1.

5 Conclusion

In this paper, we propose a model for design template generation from input text. A design template encompasses a background image and a layout of foreground elements. The core idea underlying our method is to pretrain two independent, expressive diffusion priors, one for text-to-image generation and one for unconditional layout generation, and merge them into an unified diffusion model where the two pretrained denoising networks collaborate to generate the background image and the layout jointly through a trainable communication module. Furthermore, we introduce a simple guidance technique to enforce design principles in generated results without any retraining. Our experimental results demonstrate that our model is superior to existing alternative methods, synthesizing visually aesthetic background images, high-quality layouts, and harmonious image-layout compositions.

REFERENCES

Diego Martin Arroyo, Janis Postels, and Federico Tombari. Variational transformer networks for layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13642–13652, 2021.

Yunning Cao, Ye Ma, Min Zhou, Chuanbin Liu, Hongtao Xie, Tiezheng Ge, and Yuning Jiang. Geometry aligned variational transformer for image-conditioned layout generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1561–1571, 2022.

Chin-Yi Cheng, Forrest Huang, Gang Li, and Yang Li. Play: parametrically conditioned layout generation using latent diffusion. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Yutao Cheng, Zhao Zhang, Maoke Yang, Nie Hui, Chunyuan Li, Xinglong Wu, and Jie Shao. Graphic design with large multimodal model. *arXiv preprint arXiv:2404.14368*, 2024.

Julian Jorge Andrade Guerreiro, Naoto Inoue, Kento Masui, Mayu Otani, and Hideki Nakayama. Layoutflow: flow matching for layout generation. In *European Conference on Computer Vision*, pp. 56–72. Springer, 2024.

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.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

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.

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.

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

Peidong Jia, Chenxuan Li, Zeyu Liu, Yichao Shen, Xingru Chen, Yuhui Yuan, Yinglin Zheng, Dong Chen, Ji Li, Xiaodong Xie, et al. Cole: A hierarchical generation framework for graphic design. *arXiv preprint arXiv:2311.16974*, 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.

Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Constrained graphic layout generation via latent optimization. In *ACM International Conference on Multimedia*, MM '21, pp. 88–96, 2021. doi: 10.1145/3474085.3475497.

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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pp. 491–506. Springer, 2020.

Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. Layoutgan: Generating graphic layouts with wireframe discriminators. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HJxB5sRcFQ.

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.

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 Computer Vision and Pattern Recognition Conference*, pp. 8128–8137, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Yifan Pu, Yiming Zhao, Zhicong Tang, Ruihong Yin, Haoxing Ye, Yuhui Yuan, Dong Chen, Jianmin Bao, Sirui Zhang, Yanbin Wang, et al. Art: Anonymous region transformer for variable multi-layer transparent image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7952–7962, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Mohammad Amin Shabani, Zhaowen Wang, Difan Liu, Nanxuan Zhao, Jimei Yang, and Yasutaka Furukawa. Visual layout composer: Image-vector dual diffusion model for design layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9222–9231, 2024.

Haohan Weng, Danqing Huang, Yu Qiao, Zheng Hu, Chin-Yew Lin, Tong Zhang, and C. L. Philip Chen. Desigen: A pipeline for controllable design template generation, 2024.

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

Bo Yang and Ying Cao. Order matters: Learning element ordering for graphic design generation. *ACM Trans. Graph.*, 44(4), July 2025. ISSN 0730-0301. doi: 10.1145/3730858. URL https://doi.org/10.1145/3730858.

Hui Zhang, Dexiang Hong, Maoke Yang, Yutao Chen, Zhao Zhang, Jie Shao, Xinglong Wu, Zuxuan Wu, and Yu-Gang Jiang. Creatidesign: A unified multi-conditional diffusion transformer for creative graphic design. *arXiv preprint arXiv:2505.19114*, 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.

Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson WH Lau. Content-aware generative modeling of graphic design layouts. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019.

Min Zhou, Chenchen Xu, Ye Ma, Tiezheng Ge, Yuning Jiang, and Weiwei Xu. Composition-aware graphic layout gan for visual-textual presentation designs. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 4995–5001. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/692. URL https://doi.org/10.24963/ijcai.2022/692. AI and Arts.

A Appendix

A.1 Interpretability and Effectiveness of the Communication Module

A.1.1 Training Stage Attention Map Visualization

To better understand how the layout and image modalities interact during training, we visualize cross-attention maps in both directions: from layout tokens to image patches and from image patches to layout tokens.

As shown in Figure 4, the attention from layout tokens to image features evolves significantly during training. In the early stage (step 0), each layout token attends almost uniformly to the entire image, without clear spatial preference. As training progresses, however, these attentions become sharper and more semantically meaningful: after 1000 steps, layout tokens are able to consistently focus on salient objects and object boundaries (e.g., text regions or buttons), demonstrating that the communication module learns to capture the spatial layout of the background image and align layout elements with visual context.

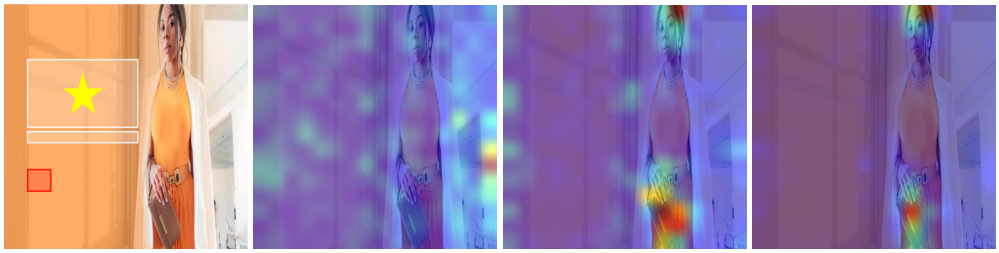
Complementary to this observation, Figure 5 shows the reverse direction of attention, from image patches to layout tokens. At the beginning of training, different image patches attend to layout tokens with little distinction, resulting in nearly uniform attention distributions. As training continues, the differences become more pronounced: some patches strongly highlight their corresponding layout tokens while others show weaker activations. This growing divergence indicates that image patches gradually specialize in recognizing their most relevant layout elements, which opens up an interesting direction for understanding how cross-modal grounding emerges during joint training.

A.1.2 Inference Stage Attention Map Visualization

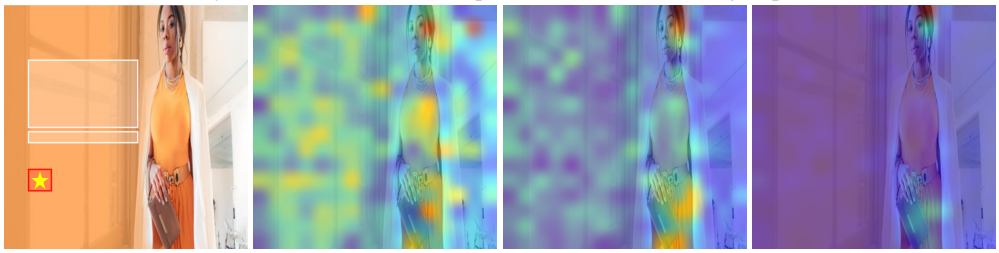
We further inspect the cross-attention weights from layout tokens to image patches during inference as the denoising process unfolds. As shown in Figure 6, even at early timesteps ($t = 701$), layout tokens already focus on salient regions of the image, such as objects and prominent anchors. As denoising progresses, these attentions sharpen and stabilize, indicating that the communication module dynamically refines its cross-modal grounding. Importantly, this behavior is consistent with the training stage observations: the communication module preserves the spatial alignment ability it learned during training and can readily attend to salient regions and boundaries in the image during inference. This consistency suggests that the learned cross-modal alignment is not only acquired in training, but also effectively retained and exploited at inference time.

A.1.3 Comparison With and Without the Communication Module

To directly examine the effect of the communication module, we compare generation results under two settings: without and with the module enabled. As shown in Figure 7, the first row corresponds to a variant where layout and image are modeled independently. In this case, there is no exchange of information and the two branches simply operate in parallel, often resulting in templates that appear disorganized and lack clear visual emphasis. The second row shows results from the same text prompts but with the communication module active. Here, the interaction between layout and image learned during training is effectively utilized at inference, allowing the two branches to guide each other. The generated templates are visibly more harmonious and better aligned, with layout elements naturally adapting to image context and vice versa. Together with the attention map analyses in the training and inference stages, This comparison highlights a consistent picture: enabling the communication module effectively builds an information bridge between the two domains, allowing layout and image to exchange complementary cues.



(a) Ground truth layout (left) and attention maps for **Token 0** at denoising steps 0, 200, and 1000.



(b) Ground truth layout (left) and attention maps for **Token 2** at training steps 0, 200, and 1000.

Figure 4: Visualization of cross-attention maps from layout tokens to image features at different training steps. In each row, the leftmost image shows the ground truth , where the element marked with a golden star indicates the selected layout token. The first row corresponds to a **text** element (white bounding box), and the second row corresponds to a **button** element (red bounding box). The subsequent images in each row show how the selected token attends to image features at training steps 0, 200, and 1000, respectively. As training progresses, attention becomes increasingly focused on salient image regions, demonstrating the communication module's ability to perceive and align with the spatial structure of the background image.

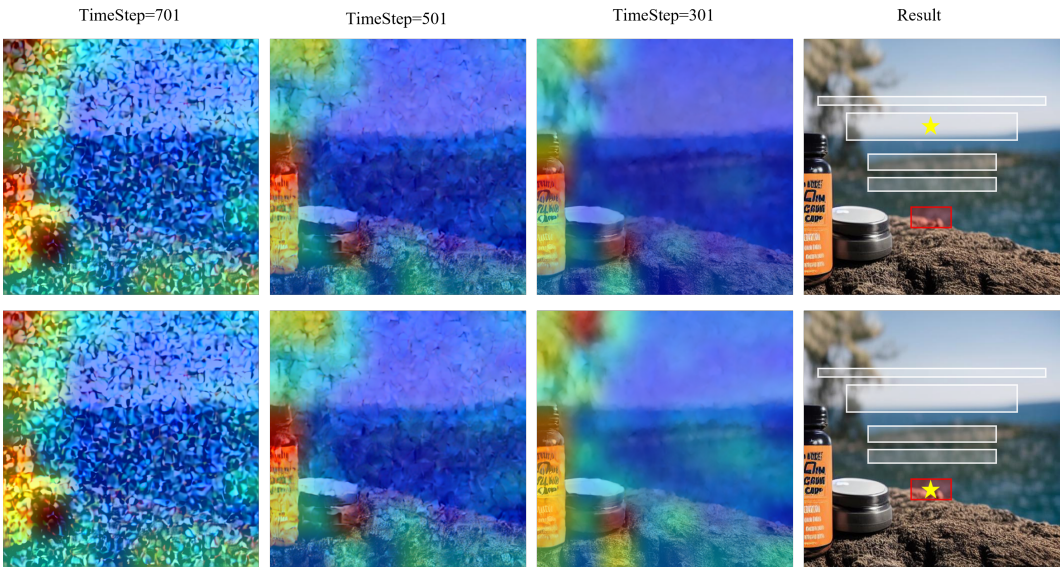


Figure 6: Visualization of cross-attention maps from layout tokens to image patches at different denoising steps ($t = 701, 501, 301$). For several representative layout tokens, we show which image regions they attend to. Even at early stages, layout tokens focus on salient regions, suggesting that the communication module effectively captures meaningful cross-modal correspondences early on.

A.2 GPT-4V Evaluation

Inspired by recent work on graphic design generation Jia et al. (2023); Inoue et al. (2024), we leverage GPT-4V to automatically assess the quality of generated design templates. Specifically, given

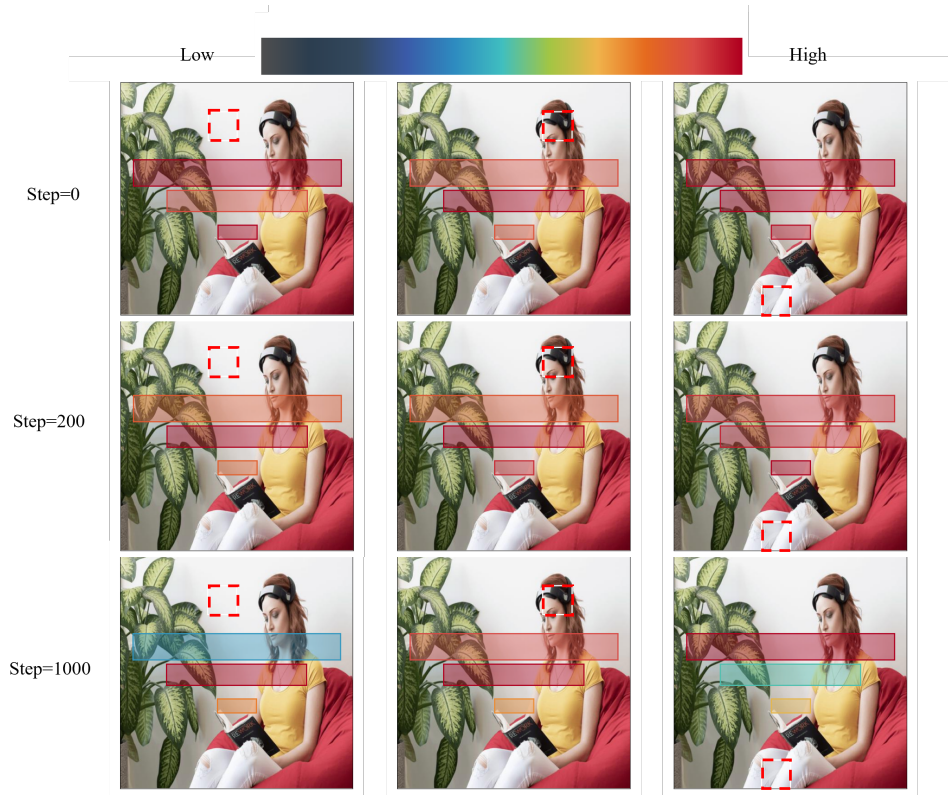


Figure 5: Cross-attention visualization from image patches to layout elements at different denoising timesteps. Each red dashed box indicates a selected image patch, and the semi-transparent color over each layout box reflects its attention weight from that patch (red = high attention, blue = low attention). The attention values are normalized and mapped to a colormap for clarity.

Figure 7: Visual comparison of generated design templates with and without the communication module. The first row shows results from the variant **without communication**, while the second row shows results **with communication**. Each column corresponds to the same input prompt across both variants.

a design template, GPT-4V is instructed to evaluate it across five aspects, each scored on a scale of 1 to 10, with higher values indicating better performance. Based on text inputs randomly sampled from the test dataset, each candidate model (Desigen, OpenCOLE, and ours) generates 2,000 design templates. The aggregated GPT-4V scores are reported in Table 4. To ensure consistency in evaluation, we design a specialized prompt to guide GPT-4V. Building on the COLE framework Jia et al. (2023), we adapt and refine the prompt to better fit the characteristics of design templates, as shown in PromptBox A.12. The five evaluation metrics capture complementary aspects of design quality:

- *Layout Structure and Readability (LSR):* Measures whether the bounding boxes form a clear and balanced layout that supports natural reading order.
- *Content Role Allocation (CRA):* Evaluates whether elements such as text, buttons, and images are used semantically in line with their intended roles.
- *Layout Adaptability and Spatial Logic (LAS):* Assesses how well the layout maintains structural consistency, including proportional alignment and robustness to content variation.
- *Visual Balance with Background (VBB):* Measures how harmoniously the layout integrates with the background image, avoiding excessive occlusion or imbalance.
- *Prompt–Background Alignment (PBA):* Evaluates whether the generated design matches the semantics of the input text prompt and whether the background is aesthetically suitable.

As summarized in Table 4, our model consistently outperforms prior methods across all five aspects. For example, our method achieves an LSR score of 7.24 compared to 6.91 for Desigen and 6.07 for OpenCOLE. In CRA, we obtain 7.42 versus 6.30 (Desigen) and 5.43 (OpenCOLE), showing a clear gain of more than +1 point over both baselines. The improvement is even more pronounced in LAS (8.62), which is +1.29 higher than Desigen and nearly +3.8 higher than OpenCOLE, demonstrating that our dual-branch communication mechanism greatly enhances layout adaptability. For VBB, our model slightly improves upon Desigen (6.65 vs. 6.59) and remains close to real data (6.76), indicating strong visual balance with the background. Finally, in PBA, our model achieves 7.64, outperforming Desigen (7.32) and OpenCOLE (7.26), showing better alignment between prompt semantics and generated designs. While there remains a small gap to real human-designed templates (e.g., 7.83 in LSR and 8.30 in CRA), these results confirm that our method produces more coherent, semantically accurate, and visually consistent design templates than existing baselines.

Method	LSR↑	CRA↑	LAS↑	VBB↑	PBA↑
Desigen	<u>6.91</u>	<u>6.30</u>	<u>7.33</u>	<u>6.59</u>	<u>7.32</u>
OpenCOLE	6.07	5.43	4.78	5.75	7.26
Ours	**7.24**	**7.42**	**8.62**	**6.65**	**7.64**
Real Data	7.83	8.30	8.48	6.76	7.98

Table 4: GPT-4V evaluation results. The bottom row shows real data. Best and second-best are bold and underlined.

A.3 Effectiveness of Occlusion Guidance

A.3.1 Step-wise Denoising Visualization

To qualitatively assess the effect of occlusion guidance during generation, we visualize the denoising trajectory of a sample under guidance scale $= 3$. As shown in Figure 8, the layout elements initially overlap with salient regions of the image (e.g., foreground objects). As denoising proceeds, however, these elements are gradually pushed away from the salient content and repositioned into less intrusive areas. This step-wise visualization clearly demonstrates that the occlusion guidance mechanism effectively drives the layout to avoid covering important visual content throughout the generation process.

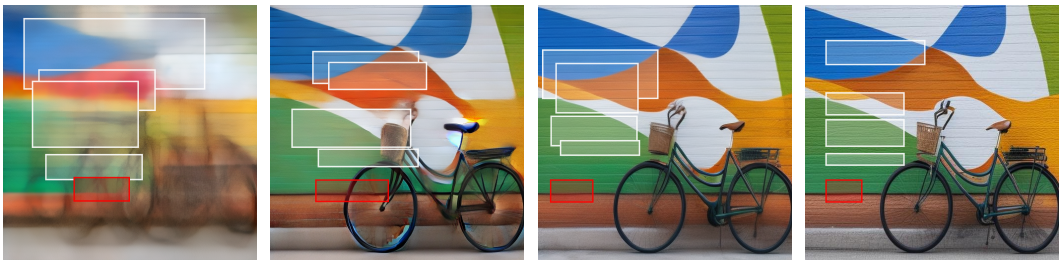
A.3.2 Ablation on Guidance Scale

We conduct an ablation study on the occlusion guidance scale s to better understand its effect on layout generation. Since this guidance directly influences only the placement of layout elements, we evaluate both layout-level and template-level metrics (Table 5).

As s increases, the model becomes more conservative in placing layout elements over salient regions. Occlusion decreases consistently and reaches its minimum at larger scales (e.g., $s = 10$), indicating that strong guidance effectively enforces avoidance of important background areas. Readability also

improves with moderate scales (around $s = 3$ to $s = 5$), but excessive values lead to diminishing returns. At the same time, stronger guidance introduces side effects: LayoutFID and alignment gradually deteriorate, and template-level realism is harmed. Specifically, TemplateFID rises steadily (e.g., 86.63 $\rightarrow$ 114.34), while TemplateCLIP drops slightly (3.25 $\rightarrow$ 3.16), suggesting that overly strong guidance distorts the natural distribution of layouts and weakens overall design quality.

A moderate guidance scale achieves the best trade-off. In particular, $s = 2$ and $s = 3$ strike a balance between reducing occlusion and maintaining competitive LayoutFID, alignment, and template-level realism. Qualitative results in Figure 9 confirm this trend: at $s = 3$, layout elements are already well separated from salient objects in the background, resulting in cleaner and more harmonious compositions without sacrificing fidelity. Therefore, we adopt $s = 3$ as the default setting in our main experiments. Larger scales (e.g., $s = 7$ or $s = 9$) can be used in applications where strict occlusion avoidance is preferred, but they may come at the cost of reduced layout fidelity and visual realism.



(a) Step 800 (b) Step 500 (c) Step 200 (d) Step 0

Figure 8: Denoising trajectory under occlusion guidance (occ scale = 3). The layout progressively moves away from salient image regions as the denoising steps proceed.

Table 5: Ablation on different Guidance scales (s) under DDIM step=50. For each metric, the best result is marked in **bold**, and the second best is underlined. The last row shows statistics computed from real data as reference.

Setting	Layout					Template	
	L-FID↓	Align	Overlap	Occlusion	Readability	TemplateFID↓	TemplateCLIP↑
s=0	**0.148**	**0.31**	11.98	**19.30%**	11.41%	86.63	3.25
s=1	0.1547	0.42	11.90	15.40%	10.50%	87.05	3.25
s=2	0.1874	0.38	**11.38**	13.40%	10.27%	89.41	3.24
s=3	0.2113	0.32	13.26	12.66%	10.35%	92.81	3.23
s=4	0.2274	0.36	11.80	12.21%	10.25%	94.35	3.23
s=5	0.2899	0.34	13.47	11.22%	10.26%	104.83	3.22
s=6	0.2946	0.41	13.06	11.22%	10.19%	104.32	3.21
s=7	0.3324	0.34	13.79	10.35%	**9.97%**	109.44	3.21
s=8	0.3658	0.48	13.58	10.07%	10.01%	112.37	3.19
s=9	0.3689	0.46	13.85	10.25%	10.22%	112.48	3.18
s=10	0.4426	0.47	13.73	9.71%	10.05%	114.34	3.16
Real Data	–	0.31	9.34	21.14%	8.53%	–	3.39

A.4 User Study on Template Embedding Similarity

To further validate the perceptual quality of our learned template embedding space, we conducted a user study involving 40 participants, including both design professionals and non-experts. Each participant was shown triplets of the form $(x, \tilde{x}_{\text{rand}}, \tilde{x}_{\text{emb}})$, where x is a reference template, $\tilde{x}_{\text{rand}}$ is a randomly selected template from the Web-design dataset, and $\tilde{x}_{\text{emb}}$ is the most similar template to x retrieved by our learned embeddings. Participants were asked to choose, based on their intuition, which of $\tilde{x}_{\text{rand}}$ or $\tilde{x}_{\text{emb}}$ is more similar to x, considering both layout and image content.

Figure 10 shows three representative cases, with the *Preference Rate* on the right indicating how often $\tilde{x}_{\text{emb}}$ was chosen over $\tilde{x}_{\text{rand}}$. In Case 1, $\tilde{x}_{\text{emb}}$ was preferred 90% of the time, in Case 2 the

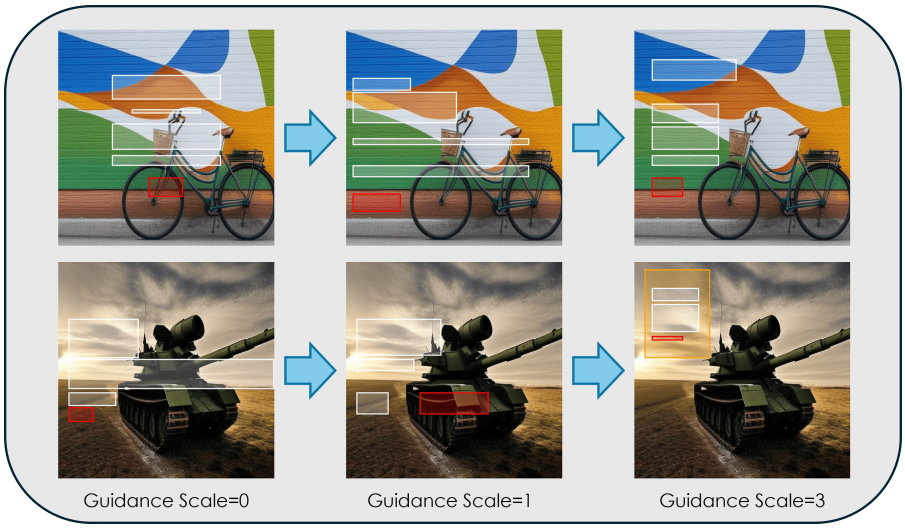


Figure 9: Visualization of two samples under different guidance scales. From left to right: guidance scale = 0, 1, and 3. The comparison shows how stronger occlusion guidance encourages layout elements to avoid salient regions more consistently, resulting in cleaner and more readable templates.

rate was 77.5%, and in Case 3 it was 72.5%. Across all 20 triplets, $\tilde{x}_{\text{emb}}$ was chosen in 73.8% of comparisons, demonstrating that our learned embeddings align well with human perception. These results also suggest that the proposed TemplateFID metric is consistent with human judgments of template similarity.

A.5 Additional Ablation: Effect of Communication Ratio η

To better understand how the choice of communication ratio η influences the denoising process, we vary the proportion of timesteps in which the communication module is active. Here, $\eta = 100\%$ means cross-attention is enabled throughout all denoising steps, while $\eta = 0\%$ disables the module entirely. Intermediate values restrict communication to a partial portion of the denoising trajectory; for example, $\eta = 30\%$ indicates that cross-modal exchange is only enabled during the first 30% of the denoising steps. Table 6 reports the results across image- and layout level metrics, as well as inference time.

Several interesting patterns emerge: (1) *Disabling communication ($\eta = 0\%$)* yields the lowest FID (19.32) and highest CLIP score (30.37), but at the cost of severe layout degradation. In particular, occlusion (30.74%) and saliency ratio (21.37%) deviate significantly from real data (21.14% and 14.17%), indicating that the model prioritizes image realism but fails to achieve layout–image coordination. (2) *Excessive communication ($\eta = 100\%$)* produces layouts that are closer to real saliency levels (14.31%), but degrades image FID (22.84) and increases overlap (15.18). This suggests that constant cross-modal exchange may introduce noise and harm image fidelity. (3) *Moderate communication ratios* (e.g., $\eta = 30\%$–50%) strike a better balance. For instance, $\eta = 30\%$ achieves near-optimal LayoutFID (0.15), the best alignment (0.31), and reduced overlap (11.98), while keeping FID (19.57) and CLIP (29.79) competitive.

Inference time also decreases slightly as η is reduced (5.24s $\rightarrow$ 4.91s), but the effect is marginal compared to the trade-offs in generation quality.

Overall, this study highlights that communication is most beneficial when applied selectively rather than continuously. We adopt $\eta = 30\%$ as the default in our main experiments, as it consistently improves layout coherence and template-level realism without sacrificing image quality.

A.6 Dataset

We conduct our experiments on the Web-design dataset Weng et al. (2024), which consists of around 50K web banner designs collected from real-world online shopping platforms. Each sample includes a high-resolution background image, structured layout annotations (bounding boxes and element types), and a product description that conveys information such as commercial purpose, theme, or

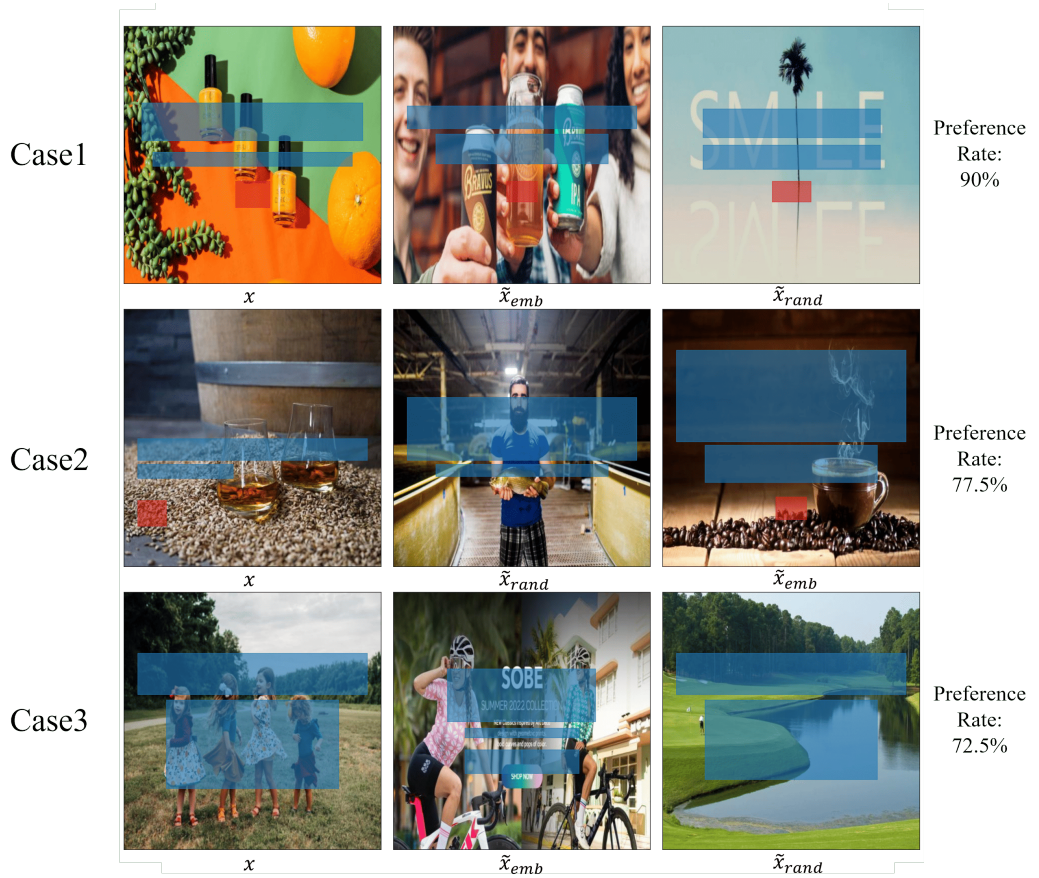


Figure 10: User study examples. Each row shows one triplet: reference x (left), embedding-based retrieval $\tilde{x}_{\text{emb}}$ (middle), and random retrieval $\tilde{x}_{\text{rand}}$ (right). The *Preference Rate* on the right shows how often $\tilde{x}_{\text{emb}}$ was chosen by participants. Blue bounding boxes denote text regions, while red bounding boxes denote button elements.

Setting	Image		Layout						Inference Time (s/template)
	FID↓	CLIP↑	Saliency Ratio	LayoutFID↓	Align	Overlap	Occlusion	Readability	
$\eta = 100\%$	22.84	27.97	**14.31%**	0.51	0.41	15.18	18.97%	13.38%	5.24
$\eta = 70\%$	22.39	28.44	14.53%	0.36	0.37	14.41	18.94%	12.79%	5.17
$\eta = 50\%$	21.66	28.42	14.34%	0.24	0.35	13.19	19.07%	12.37%	5.07
$\eta = 30\%$	19.57	29.79	14.56%	0.15	**0.31**	11.98	**19.30%**	**11.41%**	5.01
$\eta = 10\%$	19.45	29.87	19.07%	**0.14**	0.32	**11.41**	26.76%	13.27%	4.97
$\eta = 0\%$	**19.32**	**30.37**	21.37%	0.16	0.38	12.87	30.74%	12.23%	**4.91**
Real Data	–	27.5	14.17%	–	0.31	9.34	21.14%	8.53%	–

Table 6: Ablation study on different η values. For all metrics except CLIP and FID-related ones, values closer to those calculated from real data indicate better performance. The last column reports inference time per template. Best and second-best are bold and underlined, respectively.

target audience. We use 41,270 samples (85%) for training, 2,427 samples (5%) for validation, and 4,856 samples (10%) for testing.

A.7 Implementation Details of the DualFlow Model

Layout Prior. Each layout is represented as a sequence of up to 7 elements, where each element is defined by five attributes: category, left coordinate, top coordinate, width, and height. We discretize each attribute into 64 values by replacing raw coordinates with their nearest k-means cluster index. Layouts with fewer than 7 elements are padded to a fixed length. Notably, only about 0.97% of layouts in the dataset contain more than 7 elements.

Image Prior. For a fair comparison between our method, OpenCOLE, and Desigen, we use the same SD-v1.4 backbone and set the output resolution to 512×512, consistent with Desigen (see Figure 3). For the teaser and additional qualitative results in the supplementary material, we adopt SD-v2.0 with an output resolution of 768×768 (see Figure 1 and Figure 12).

We use a β-VAE to model the layout, with a latent size of 32. The VAE encodes layout tokens into compact latent representations. For layout denoising, we adopt a transformer-based architecture using DiT as the backbone. Specifically, our Layout LDM consists of 28 transformer blocks, each with 8 attention heads and a query/key/value dimension of 512. This configuration enables the model to effectively capture spatial dependencies and hierarchical structures across layout elements during the denoising process.

Communication Module. In the joint modeling stage, we introduce a communication module that facilitates feature exchange between the layout and image backbones. Specifically, we extract multi-scale image features from the second and third downsampling blocks as well as the middle block of the image U-Net, and layout features from the 20th transformer block of the layout diffusion model. Cross-attention is performed in both directions: layout features attend to image features to incorporate visual context, while image features attend to layout representations to integrate structural information. The attended features are then fused back into their respective backbones through residual connections. During this stage, both the pre-trained layout diffusion model and image backbone are kept frozen, and only the communication module is updated.

Inference Details. During inference, we use a DDIM sampler for the layout backbone and a DDPM sampler for the image backbone, each with 50 sampling steps. The communication module is enabled only after step 700 (corresponding to $\eta = 30\%$ of the denoising trajectory) to facilitate cross-modal information exchange. In addition, occlusion-avoidance guidance is applied between steps 200 and 800 to steer layout elements away from salient background regions.

A.8 Training Details

Our training pipeline consists of two main stages: (1) **Building Domain-specific Priors**, which includes pretraining the layout VAE, training the Layout LDM, and fine-tuning Stable Diffusion; and (2) **Training the Joint Model**, which focuses on learning the cross-domain communication module. All experiments are conducted on an NVIDIA A40 GPU with 48GB memory.

Stage 1: Building Domain-specific Priors.

- *Layout VAE Training.* We employ a β-VAE Higgins et al. (2017) to encode layouts into latent representations. The β coefficient is set to 5×10^{-4} to find a sweet spot between reconstruction accuracy and latent disentanglement.
- *Layout LDM Training.* The layout diffusion model is trained to model the distribution of latent layouts. We use a batch size of 4096 and train for 1000 epochs with a learning rate of 1×10^{-4}.
- *Stable Diffusion Fine-tuning.* To ensure fair comparison with **Desigen** Weng et al. (2024), the previous state-of-the-art method, we fine-tune the Stable Diffusion model on the corresponding dataset. The model is trained for 100 epochs with a batch size of 8 and a learning rate of 1×10^{-5}.

Stage 2: Training the Joint Model.

- *Cross-Domain Communication Module.* The feature exchange module is trained independently to facilitate information transfer between the layout and image domains. The model is trained for 20 epochs with a batch size of 20 and a learning rate of 1×10^{-4}.

Optimization. All training stages adopt the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 1×10^{-2}. The learning rate schedule follows a cosine decay strategy.

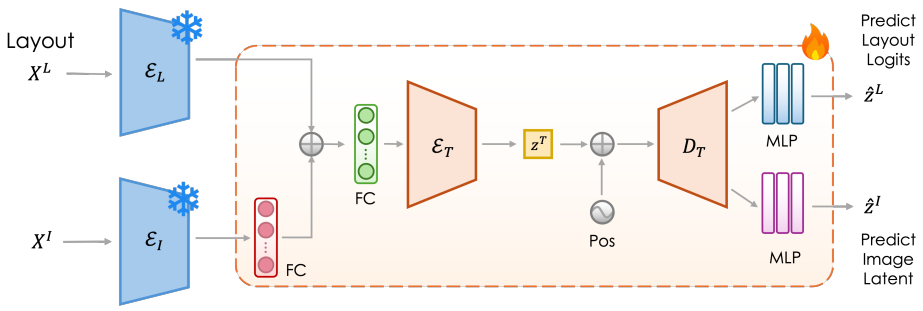


Figure 11: Architecture of the template autoencoder (TemplateAE). The frozen domain-specific encoders $\mathcal{E}_\text{I}$ and $\mathcal{E}_\text{L}$ map image and layout to latent space. The concatenated embeddings are processed through template encoder $\mathcal{E}^T$ to produce unified design latent z^T, which is then decoded by template decoder $\mathcal{D}^T$ and two MLPs to reconstruct image latent and layout logits.

A.9 Implementation Details of Occlusion Avoidance Guidance

We define a loss function for occlusion avoidance to guide the layout generaition process. Specifically, given the predicted image noise $\hat{\epsilon}_t^I$, we compute a clean image latent $\hat{z}_0^I$ analytically using the forward process margin distribution $q(z_t^I|z_0^I)$, decode it into an image $\mathcal{D}_\text{I}(\hat{z}_0^L)$, and compute a saliency map $S = f_\text{sal}(\mathcal{D}_\text{I}(\hat{z}_0^L))$ using an off-the-shelf saliency detector f_sal. Then, we compute a clean layout latent $\hat{z}_0^L$ from the predicted layout noise $\hat{\epsilon}_t^L$ with $q(z_t^L|z_0^L)$, and decode it into a layout $\mathcal{D}_\text{I}(\hat{z}_0^L)$. The occlusion avoidance loss $\mathcal{L}_\text{occ}$ is computed as the average saliency value within the decoded layout elements on S. Let $\{\mathbf{b}_i\}_{i=1}^B$ be a set of decoded element bounding boxes. $\mathcal{L}_\text{occ}$ is written as:

$$\mathcal{L}_\text{occ} = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{A_i}M_{\mathbf{b}_i} \odot S, \tag{8}$$

where $M_{\mathbf{b}_i}$ is a soft mask for $\mathbf{b}_i$ with higher values at locations within $\mathbf{b}_i$, and A_i is the area of $\mathbf{b}_i$. Since the decoded layout elements has discrete bounding box parameters, to make $\mathcal{L}_\text{occ}$ differentiable, we convert them into continuous ones by estimating each continuous parameter as a weighted sum of all quantization bin centers for the parameter, where the weights are predicted probabilities over all the bins. We denote by $\mathbf{b}_i = (x_i, y_i, w_i, h_i) \in \mathbb{R}^4$ the estimated continuous bounding box parameters of i-th element, where (x_i, y_i) denotes the top-left location, and w_i and h_i are, respectively, the width and height. From $\mathbf{b}_i$, we compute the top-left coordinates (x_i^l, y_i^t) and the bottom-right coordinates (x_i^r, y_i^b) of the bounding box. The soft mask $M_{\mathbf{b}_i}$ is computed in differentiable manner, with the value at position (x, y):

$$\begin{aligned} M_{\mathbf{b}_i}(x,y) = \sigma(\lambda(x - x_i^l)) \times \sigma(\lambda(x_i^r - x)) \\ \times \sigma(\lambda(y - y_i^t)) \times \sigma(\lambda(y_i^b - y)), \end{aligned} \tag{9}$$

where $\sigma(\cdot)$ is the sigmoid function. λ is set to 40.

A.10 TemplateAE Architecture Details

To enable holistic evaluation of design templates, we train a template autoencoder (TemplateAE) to extract joint image-layout embeddings. As shown in Figure 11, given a background image X^I and layout X^L, they are first mapped to latent space through frozen domain-specific encoders $\mathcal{E}_\text{I}$ and $\mathcal{E}_\text{L}$ to obtain $z^I = \mathcal{E}_\text{I}(X^I)$ and $z^L = \mathcal{E}_\text{L}(X^L)$. These embeddings are concatenated and processed through a template encoder $\mathcal{E}^T$ to produce a unified design latent z^T. Subsequently, z^T is combined with cosine position embeddings and fed into a template decoder $\mathcal{D}^T$, followed by two separate MLPs that reconstruct the respective domain outputs: image latent $\hat{z}^I$ and layout logits $\hat{z}^L \in \mathbb{R}^{N\times d^L \times V}$. During training, we freeze $\mathcal{E}_\text{I}$ and $\mathcal{E}_\text{L}$ and only optimize $\mathcal{E}^T$, $\mathcal{D}^T$, and the MLPs. The training objective on the Web-design train split combines reconstruction losses for both modalities:

$$\mathcal{L}_\text{rec} = \gamma\big(1 - \frac{z^I \cdot \hat{z}^I}{\|z^I\|\|\hat{z}^I\|}\big) + \sum_{i=1}^{N}\sum_{j=1}^{V} -X_{i,j}^L \log(\hat{Z}_{i,j}^L) \tag{10}$$

where the first term measures image latent reconstruction quality and the second term evaluates layout prediction accuracy. γ is the balance factor that makes the two terms in comparable scale, in our experiment we choose $\gamma = 5.0$.

A.11 Additional Results

Figure 12 presents additional design templates generated by our model. The first three rows correspond to results produced *without* applying the occlusion-aware guidance (OAG), while the last two rows show results generated by our model *with* OAG enabled using a guidance scale of 3. This comparison illustrates how the proposed guidance further improves spatial arrangement by steering layout elements away from salient regions in the background, leading to cleaner and more readable templates.

A.12 GPT-4V-Based Evaluation Protocol for Design Templates

Quality Assurance Prompt for GPT-4V (Vision)

You are an autonomous AI Assistant who aids designers by providing insightful, objective, and constructive critiques of graphic design templates.
Your goals are: Deliver comprehensive and unbiased evaluations of design templates based on established layout principles and UI/UX best practices. Identify potential areas for improvement and suggest actionable feedback to enhance the overall usability and visual clarity of the template. Maintain a consistent and high standard of critique.
Utilize coordinate information for data description relative to the upper-left corner of the image, with the upper-left corner serving as the origin, the right as the positive x-axis, and downward as the positive y-axis.
Please abide by the following rules: Strive to score as objectively as possible. Grade seriously. A flawless design template can earn 10 points, a mediocre one can only earn 7 points, a template with obvious shortcomings can only earn 4 points, and a very poor template can only earn 1–2 points. Keep your reasoning concise when rating, and describe it as briefly as possible.
Grading criteria:

- **Layout Structure and Readability (1–10):** The layout should be logically organized and visually balanced. Bounding boxes should follow a coherent visual hierarchy, avoid overlap or crowding, and support natural reading flow. A 10 indicates a clean and clear layout that maximizes clarity and usability; a 1 indicates poor structure, chaotic arrangement, or unreadable positioning.
- **Content Role Allocation and Semantics (1–10):** The semantic use of bounding boxes should match their intended UI/UX roles. A score of 10 means perfect role placement and function; a 1 indicates confusing or inappropriate usage.
- **Layout Adaptability and Spatial Logic (1–10):** Bounding boxes should be well-aligned, proportionally distributed, and adaptable to different content lengths or screen sizes. A score of 10 indicates a layout with strong structural consistency and high reusability; a 1 reflects disorganized, rigid, or easily breakable layouts.
- **Visual Balance and Composition with Background (1–10):** The layout and background image should be harmoniously integrated. The background must not cause excessive occlusion of key layout elements and should support overall visual clarity. A score of 10 indicates strong compositional balance and minimal interference; a score of 1 reflects disruptive overlap or poor visual coordination.
- **Prompt Alignment and Background Aesthetics (1–10):** The generated template should accurately reflect the content and intent of the input prompt, including not just the layout and elements but also the aesthetic quality and appropriateness of the background image. A score of 10 indicates that the design closely aligns with the user's expectations and prompt semantics, and the background is visually pleasing and well-integrated; a score of 1 reflects a major mismatch with the prompt or a background that is distracting, irrelevant, or poorly composed.

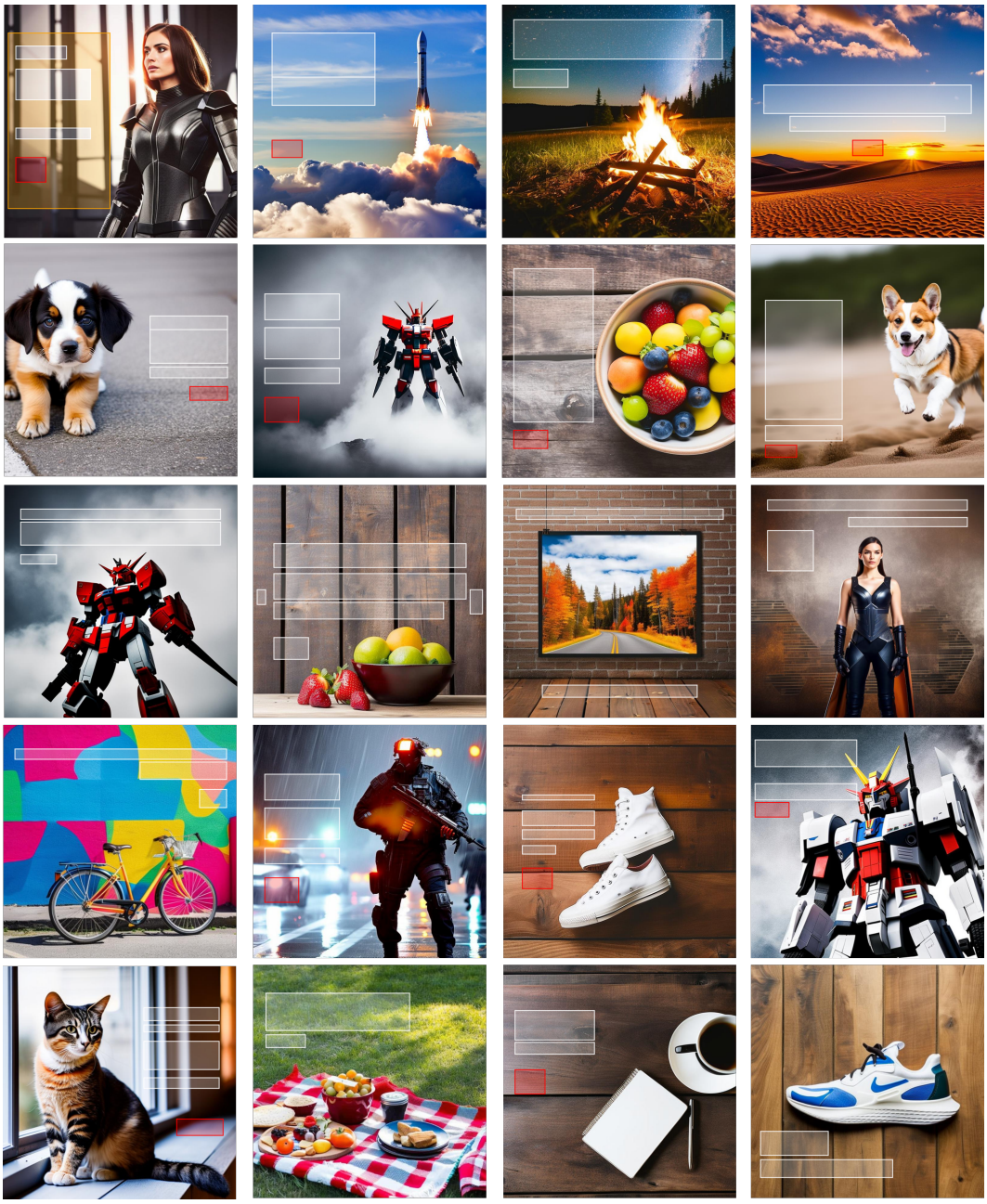


Figure 12: Additional design template samples generated by our model. Each result consists of a synthesized background image and an overlaid layout composed of multiple element categories. Bounding box colors indicate element types: red boxes denote *buttons*, white boxes denote *text*, and yellow boxes denote *underlays*. These visualizations demonstrate the model's ability to jointly generate coherent backgrounds and structured layouts with semantically meaningful role allocations and clear spatial organization.