The Visual Computer (2025) 41:4239–4252
https://doi.org/10.1007/s00371-024-03657-z

RESEARCH

Image-aware layout generation with user constraints for poster design

Chenchen Xu[1,2] · **Kaixin Han**[2] · **Weiwei Xu**[1,2]

Accepted: 11 September 2024 / Published online: 26 September 2024

Abstract

Graphic layout is essential in poster generation. Professionals often need to design different layouts for a product image, to ensure they meet specific user requirements. This paper focuses on utilizing a deep-learning model to automatically generate image-aware layouts with user-defined constraints, including layout attributes and partial layouts. Layout attribute constraints require generated layouts to include and exclude elements of specified classes, such as text, logos, underlays, and embellishments. Our model represents different attributes by sampling multidimensional Gaussian noise with different means, and we propose an attribute-consistent loss and an attribute-disentangled loss to ensure that the generated layout satisfies the specified attribute. Partial layout constraints provide our model with incomplete layout information to guide the generation of the remaining elements. We design a partial-constraint loss to incorporate the provided partial layout. Furthermore, we introduce a random mask to diversify the partial layout constraints, which can encourage the model to learn more general latent representations of the provided partial layouts. Both quantitative and qualitative evaluations demonstrate that our model can generate different image-aware layouts according to various user constraints while achieving state-of-the-art performance.

Keywords Graphic layout · Poster · Image-aware · User constraints · Layout attributes · Partial layouts

1 Introduction

Graphic layout design, which involves arranging texts, logos, underlays, and other 2D elements [1, 2], is an essential component for various media, such as magazines [3–7], posters [8–11], web pages [12–15], and comics [16–19]. The well-crafted graphic layouts are heavily reliant on the designers' experience and proficiency.

In the past decade, deep-learning-based methods for graphic layout generation have emerged [20–23]. Recently, some image-aware methods have been proposed to model the relationship between image content and graphic layout elements [2, 24–28]. However, these models are not well designed to handle user constraints that express diverse design demands. In this paper, we focus on four classes of elements, text, underlay, logo, and embellishment, and divide user constraints into two main categories: *layout attribute* and *partial layout* constraints. Layout attribute constraints are used to control that the generated layout includes the elements of required classes (attribute elements) and excludes the elements of undesired classes (undesired elements). For example, when the attribute is "layout with logos but without embellishments," generated layouts need to display the product logo without any embellishments. Partial layout constraints require the model to supplement the given incomplete layout and generate a complete layout. Although CGL-GAN [2] and PDA-GAN [25] allowed to guide the layout generation with user-specified coordinates and classes of partial elements, they do not always conform to such constraints and fail to handle the constraints with *incomplete element information*, for instance, coordinate or class only. More importantly, these models cannot handle layout attribute constraints.

This paper focuses on generating different high-quality graphic layouts according to user constraints for one product image. To this end, our proposed network integrates the layout attribute and partial layout constraints into image-aware layout generation methods, abbreviated as the IUC-Layout network. As a result, it is controllable, enabling the designer to express the diverse presentation requirements of advertising posters in the layout generation. Our model samples

✉ Weiwei Xu
xww@cad.zju.edu.cn

[1] State Key Lab of CAD and CG, Zhejiang University, Hangzhou, China

[2] College of Computer Science and Technology, Zhejiang University, Hangzhou, China

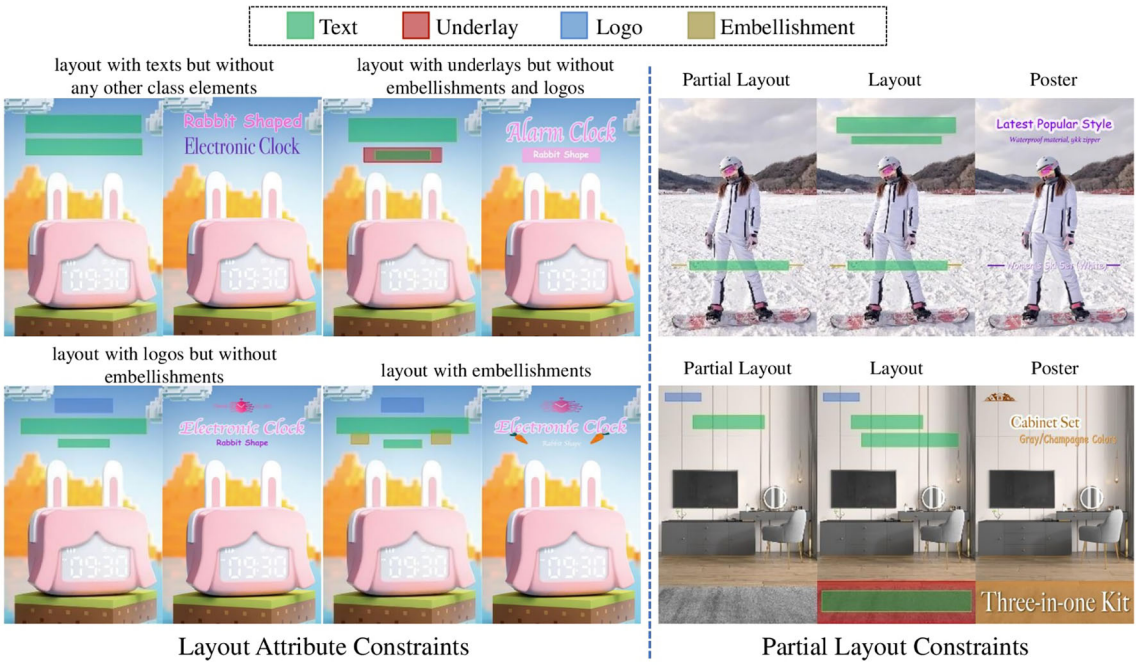


Fig. 1 Examples of generated layouts and posters with image contents and user constraints. Our model generates image-aware layouts that adhere to layout attribute constraints (left) and partial layout constraints (right), which can be used to generate advertising posters

multidimensional Gaussian noise with different means to represent different types of layout attribute constraints. Since this representation assigns each layout attribute constraint with a region, it can fill the empty region between different means with more training examples and force the network to learn the intrinsic representation of different attributes robust to the noise perturbation. We found that it is beneficial to improve the robustness of our model during training. Specifically, we sample 4 dimensional Gaussian noises to represent 4 types of layout attribute constraints. In addition, we design attribute-consistent loss and attribute-disentangled loss to ensure the layout generated by the IUC-Layout network satisfies the corresponding layout attribute constraint. They are achieved by approximately counting the number of attribute or undesired elements using softmax operation to facilitate the gradient backpropagation. As shown in Fig. 1, the graphic layout design by the model can arrange four classes of elements, including texts, underlays, logos, and embellishments, at the appropriate positions based on product images and user-specified layout attributes or partial layouts. When the layout attribute constraint is "layout with text but without any other class elements," as shown in the top-left of Fig. 1, the generated layout consists of text elements only.

Additionally, we introduce a random mask operation to obtain incomplete element information constraints for the partial layout, which can encourage our model to learn more general latent representations of provided partial layouts. We also propose a partial-constraint loss to guide models to generate layouts that are precisely consistent with the given information. In our experiments, we also integrate the partial-constraint loss and the random mask into other image-aware layout networks to further verify the benefits of these two operations when handling partial layout constraints. We summarize the contributions of this paper as follows:

- We design an efficient representation of layout attribute constraints, which can force the network to learn the intrinsic representation of different attributes robust to the noise perturbation. Two losses, attribute-consistent loss and attribute-disentangled loss, are designed to ensure that the generated layouts by IUC-Layout network satisfy user-specified attributes.
- We design a partial layout loss that guides the model to complete layouts based on the given information. Furthermore, we introduce a random mask operation to obtain incomplete element information constraints for the partial layout, to enhance the model's general latent representations.
- Both quantitative and qualitative evaluations demonstrate that our model can generate different high-quality layouts according to one product image with various user constraints while achieving SOTA performances.

2 Related works

Continuous research efforts [2, 25, 26, 29–32] have been devoted to the graphic layout generation, which can be divided into two categories based on their consideration of image content: image-agnostic and image-aware layout generation.

2.1 Image-agnostic layout generation

Early works [29, 33–36] mainly utilize templates or heuristic rules to design graphic layouts and often fail to produce flexible and various layouts. Recently, an increasing number of deep-learning-based models have been developed for generating graphic layouts [1, 22, 37–43]. LayoutGAN [1], LayoutVAE [20], and LayoutVTN [21] generate layouts from noise without any conditions. To meet the diverse user demands in real-world applications, several conditional methods [44–49] have been proposed to guide the layout generation process. The condition includes graphic layout element types, numbers, sizes, and locations. For example, AttributeGAN [41] incorporates elements' aspect ratio and location as conditions to generate graphic layouts. LayoutFormer++ [46] utilizes sequence-based control mechanisms to facilitate flexible and varied layout generation. However, the aforementioned methods primarily concentrate on modeling the internal relationships among graphic layout elements, while neglecting the connection between the graphic layout and the image content.

2.2 Image-aware layout generation

ContentGAN [50] combines visual information to generate layouts for magazine pages, but it cannot fully capture the image content as global pooling is applied to feature maps. To comprehend the visual-texture content of the image, CGL-GAN [2] and PDA-GAN [25] combine CNN and transformer [51] to synthesize image-aware graphic layouts for posters, which are the most relevant works in this discipline. Recently, more image-aware layout generation methods have been proposed for modeling the relationship between graphic layouts and image contents [24, 26, 52, 53]. However, none of these methods devote to study image-aware layout generation with layout attributes and partial layout constraints. Although CGL-GAN and PDA-GAN mentioned that they could generate layouts according to partial layout constraints to some extent, conducting numerous qualitative evaluations and observations demonstrate that generated layouts may not always conform to user constraints. Additionally, this partial layout is limited to requiring complete element information, including both class and coordinate. More importantly, these models are unable to express layout attributes.

In real-world applications, it is typically necessary to design multiple layouts based on one product image to meet various user demands for the effective presentation of advertising posters. This paper focuses on leveraging layout attributes and partial layout constraints to generate image-aware graphic layouts.

3 Method

As illustrated in Fig. 2, the structure of IUC-Layout backbone network follows the design of DETR [54], which includes a multi-scale convolutional neural network (CNN) [55, 56], a transformer encoder-decoder [51], and two fully connected layers. Firstly, the multi-scale CNN extracts image feature maps. Next, the sampled four-channel Gaussian noise according to the input layout attribute is connected with the feature maps at the final layer of the CNN, and sent to the transformer encoder for the embedding. For partial layout constraints, we encode the information of each element, i.e., its position and class, into a feature vector with the same dimension as the learned queries in DETR. Afterward, these vectors are added to the queries of the transformer decoder to guide the layout generation. In this manner, the partial layout constraint can be disabled by setting the element feature vectors to 0. Our system allows the user to specify no more than 10 elements in a partial layout. Finally, two fully connected layers respectively predict classes and bounding boxes of elements.

3.1 Layout attribute

According to the factors involved in the graphical design, such as layout styles, relationships between elements and user requirements, and the corresponding statistics of CGL-Dataset, we define four types of layout attribute constraints as follows: (1) layout with texts but without any other class elements, (2) layout with underlays but without embellishments and logos, (3) layout with logos but without embellishments, and (4) layout with embellishments. Each layout attribute includes the attribute element and non-desirable elements, as illustrated in Table 1.

The distinctions in layout attributes arise from elements' classes, but the number and arrangement of elements can vary. Similarly to [57], it is necessary to use multidimensional noise to represent the layout attribute since this representation assigns layout attribute with a region. Moreover, sampling noise in the spatial domain can improve the model's generalization capability and fault tolerance, enhancing the robustness of the model during training.

As indicated in Table 2, our model preassigns four sets of four-channel Gaussian noise corresponding to aforementioned layout attributes, along with one set for the unspecified

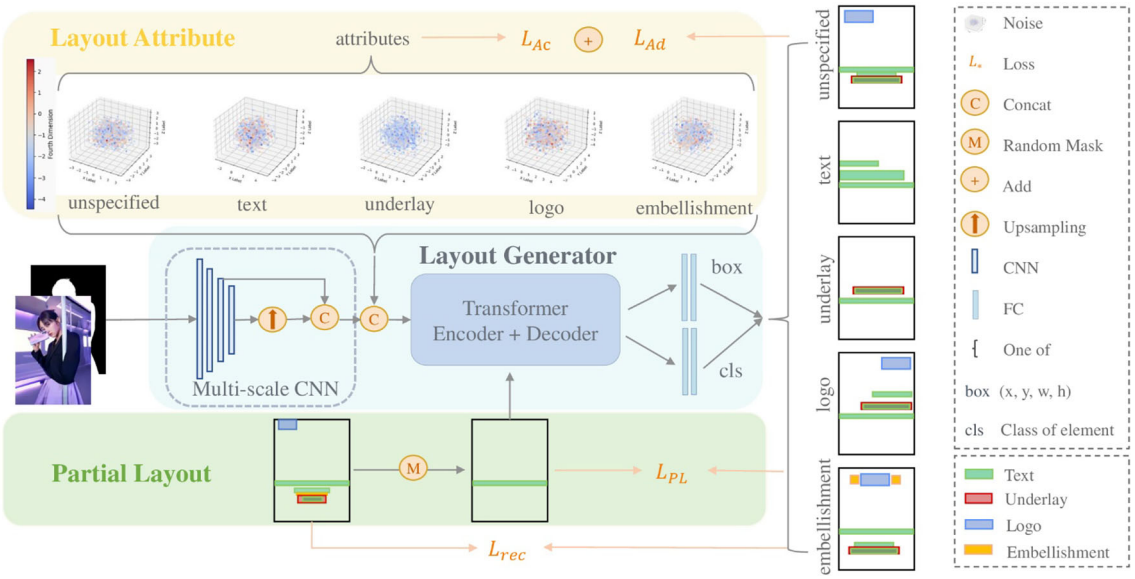


Fig. 2 The architecture of our network. The three-dimensional views along with the color map visualize the sampled 4 dimensional Gaussian noises. During each training step, our model samples noise according to the specified attribute and combines it with image contents and the partial layout to generate an image-aware layout that satisfies user constraints

Table 1 a and S^* respectively represent the attribute element and the sets of undesired elements for the specified attribute $*$

Layout attribute ($*$)	Attribute element (a)	Set of undesired classes (S^*)
Layout with texts but without any other class elements	Text	{*underlay, logo, embellishment*}
Layout with underlays but without embellishments and logos	Underlay	{*logo, embellishment*}
Layout with logos but without embellishments	Logo	{*embellishment*}
Layout with embellishments	Embellishment	{}

Table 2 Five sets of four-channel Gaussian noise with varying means based on different layout attributes

Layout attribute	Means of four-channel noise
Layout with texts but without any other class elements	(1, − 1, − 1, 1)
Layout with underlays but without embellishments and logos	(1, − 1, 1, − 1)
Layout with logos but without embellishments	(1, 1, − 1, − 1)
Layout with embellishments	(1, 1, 1, 1)
Unspecified layout attribute	(0, 0, 0, 0)

attribute. The mean values of these five sets of four-channel Gaussian noise are (1, − 1 − 1, 1), (1, − 1, 1, − 1), (1, 1, − 1, − 1), (1, 1, 1, 1), and (0, 0, 0, 0) with a variance of 1 for each channel. These points in four-dimensional space are equidistant from each other and from the origin (0, 0, 0, 0). This design effectively balances the model's learning of different layout attribute constraints. When a user specifies a certain attribute constraint, the model samples Gaussian noise in four-dimensional space according to the corresponding noise mean and variance. The length and width of each dimension of the noise vector are equal to the input feature map size of the transformer module.

To better align with the attribute element and disentangle different attributes, we design attribute-consistent loss and attribute-disentangled loss to ensure the generated layout from the corresponding Gaussian noise satisfies the specified layout attribute constraint.

3.2 Layout attribute-consistent loss

We propose an attribute-consistent loss such that the generated layout contains the attribute element. Specifically, to make the attribute-consistent loss differential, we design a

modified softmax function to approximately count the element number of each class c:

$$N_c = \sum_{q=1}^{Q} \frac{e^{z_q^c \cdot \varepsilon}}{\sum_{k \in \mathcal{K}} e^{z_q^k \cdot \varepsilon}} \tag{1}$$

where Q is the total number of output elements, and z_q^c represents the output class c of q_{th} element. The hyperparameter ε is used to increase the distinction between predicted probabilities of different classes. We set the value of ε as 100 during the training process. $\mathcal{K}$ is {*text, logo, underlay, embellishment, none*}. Therefore, we can calculate the layout attribute-consistent loss as:

$$L_{Ac} = \max(1 - N_a, 0) \tag{2}$$

where N_a is the number of attribute element a, computed by the Eq. (1). If the number of attribute elements exceeds 1, L_{Ac} is 0. Otherwise, L_{Ac} equals $(1 - N_a)$.

3.3 Layout attribute-disentangled loss

If only introducing L_{Ac}, generated layouts may contain undesired elements. To satisfy different attribute constraints, we design attribute-disentangled losses to separate the relationships between various classes. They can be formulated uniformly as:

$$L_{Ad}^{*} = \sum_{u \in \mathcal{S}^{*}} N_u \tag{3}$$

where u represents the undesired element. S^{*}, as indicated in Table 1 represents the set of classes that are absent in generated layouts based on the specified attribute $*$.

3.4 Partial layout

The partial layout constraints can be divided into two categories: one consists of elements with complete information, and the other contains elements with incomplete information. The reason to integrate elements of incomplete information is to enable the flexibility and diversity of partial layout constraints. For example, users may provide the elements with complete information or position only, or the mix of elements with complete information and elements of position information, etc. In this section, we introduce the partial-constraint loss L_P and a random mask PL_{rm} operation to obtain training examples of elements with incomplete information.

The L_P is used to train the network to produce layouts consistent with the input partial layout constraints. It is formulated as:

$$L_P = \mid \text{Pred} \cdot PL_{bm} - PL \mid \tag{4}$$

where *Pred* denotes the output of the layout generation model, and PL_{bm} is the binary matrix derived from the input partial layout PL. When a value in PL is nonzero, the corresponding entry in PL_{bm} is set to 1; otherwise, it is set to 0. The L1 distance is employed to calculate the value of L_P. The element correspondence between *pred* and PL is defined by the query index. Specifically, since we add the feature vector of the first element in a partial layout to the first query, that element should then correspond to the element produced by the first query. The rest element correspondences are done in the same way.

To augment the training with incomplete element information, we generate a random mask PL_{rm} with the same size as the partial layout, consisting of 0 and 1. The percentage of value 0 amounts to 25%. PL_{rm} randomly masked the information of the element class and coordinates in the partial layout. Similar to Eq. (5), we can calculate the loss of partial layout constraints with the random mask as follows:

$$L_{PL_{rm}} = \mid \text{Pred} \cdot PL_{bm} \cdot PL_{rm} - PL \cdot PL_{rm} \mid \tag{5}$$

The proposed partial-constraint loss and random mask are simple and can be easily applied to other models. In the experimental section, we will demonstrate their effectiveness by incorporating them into CGL-GAN and PDA-GAN.

4 Experiments

This section primarily compares our model with SOTA layout generation methods and presents its ablation studies. Given space limitations, more experimental comparisons, including a user study and analyses of computational complexity for different models, can be found in the supplementary material.

4.1 Implementation details

We implement our model in PyTorch 1.7.1 and utilize the Adam optimizer [58] for training. Initial learning rates are set to 10^{-5} for CNN, and 10^{-4} for the transformer and fully connected layers. We take CGL-Dataset as both training and test datasets. To ensure fair comparisons, following CGL-GAN and PDA-GAN, we resize the input image to 240 × 350. Our model is trained for 300 epochs with a batch size of 128. Learning rates are reduced by a factor of 10 after 200 epochs. The total training time is approximately 37.8 h, utilizing 16 NVIDIA V100 GPUs.

During training, we combine four loss functions: L_{rec}, L_{Ac}, L_{Ad}^{*} and $L_{PL_{rm}}$, to guide the model optimization. R_{rec} is the reconstruction loss that penalizes the deviation between the ground truth and the generated layout. We calculate L_{rec} following [54]. The overall training loss for the model can

Table 3 Quantitative evaluation for content-aware methods

Model	*Attribute*	$R_{lac}\uparrow$	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$	$R_{occ}\uparrow$
ContentGAN [50]	*None*	×	45.59	17.08	1.143	0.0397	0.8626	0.0071	93.4
CGL-GAN [2]	*None*	×	35.77	15.47	0.805	0.0233	0.9359	0.0098	99.6
PDA-GAN [25]	*None*	×	33.55	12.77	0.688	0.0290	**0.9481**	0.0105	99.7
IUC-Layout (Ours)	*Text*	0.973	34.23	**10.43**	**0.664**	**0.0129**	–	0.0084	97.2
IUC-Layout (Ours)	*Underlay*	0.980	**32.69**	16.64	0.816	0.0172	0.9312	**0.0030**	99.8
IUC-Layout (Ours)	*Logo*	1.000	35.93	16.27	0.936	0.0255	0.9226	0.0144	**100.0**
IUC-Layout (Ours)	*Embellishment*	0.996	33.79	15.66	0.899	0.0291	0.9163	0.0081	**100.0**
IUC-Layout (Ours)	*Unspecified*	–	33.06	15.93	0.826	0.0174	0.9221	0.0055	99.9

Bold numbers denote the best result. ↓ (or ↑) means the smaller (or bigger) value, the better. *None* means the model lacks attribute control ability. *Text* as a sample refers to the attribute "layout with texts but without any other class elements." *Unspecified* means unspecified layout attributes. × represents that the model cannot complete the corresponding task, while – indicates that the model does not need to be tested for the metric

Fig. 3 Qualitative evaluation for image-aware models. The layouts in each row are conditioned on the same product image, while the ones in a column are generated by the same model. Ours-Unsp represents unspecified attributes in our model

Table 4 Quantitative evaluation for image-agnostic methods

Model	Attribute	$R_{lac}\uparrow$	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
LayoutTransformer[22]	None	×	40.92	21.08	1.310	0.0156	0.9516	0.0049
LayoutVTN [21]	None	×	41.77	22.21	1.323	0.0130	**0.9698**	0.0047
IUC-Layout (Ours)	Text	0.973	34.23	**10.43**	**0.664**	**0.0129**	–	0.0084
IUC-Layout (Ours)	Underlay	0.980	**32.69**	16.64	0.816	0.0172	0.9312	**0.0030**
IUC-Layout (Ours)	Logo	1.000	35.93	16.27	0.936	0.0255	0.9226	0.0144
IUC-Layout (Ours)	Embellishment	0.996	33.79	15.66	0.899	0.0291	0.9163	0.0081
IUC-Layout (Ours)	Unspecified	–	33.06	15.93	0.826	0.0174	0.9221	0.0055

Bold numbers denote the best result

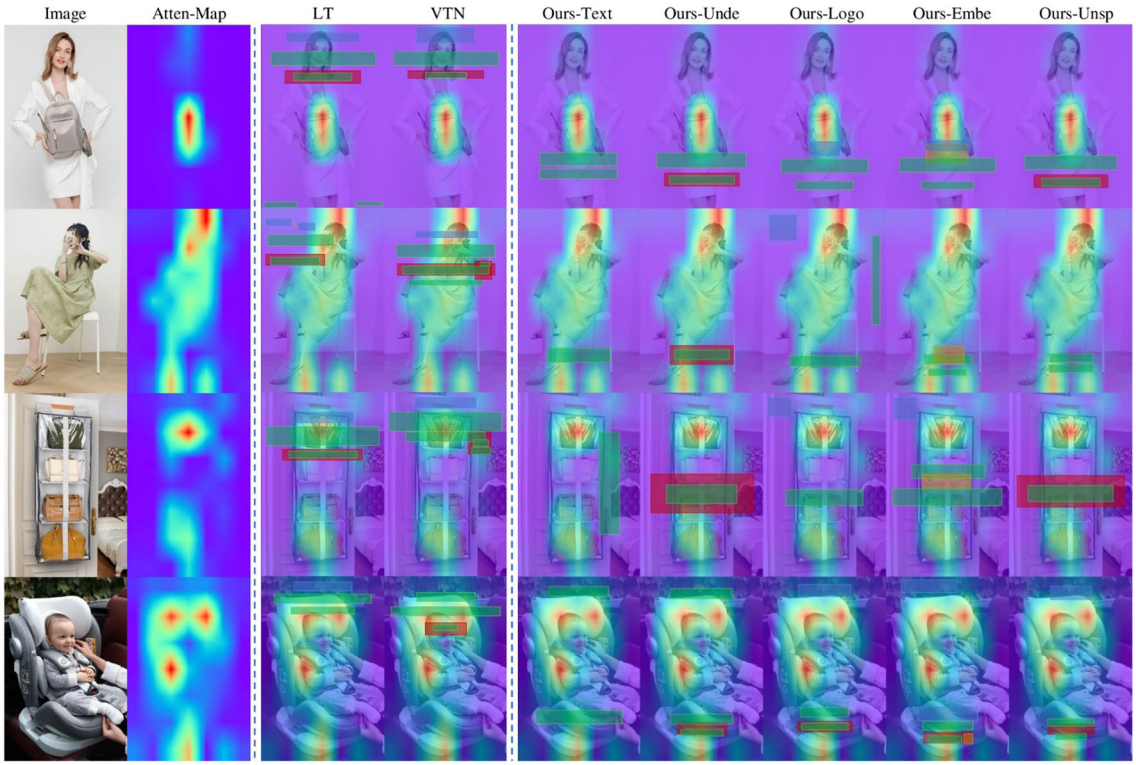


Fig. 4 Qualitative evaluation for image-agnostic models. Layouts in each row are conditioned on the same image with product attention map Atten-Map [59, 60]. LT and VTN represent LayoutTransformer and LayoutVTN, respectively

be summed as follows:

$$L = L_{rec} + \beta \cdot L_{Ac} + \gamma \cdot L^{*}_{Ad} + \eta \cdot L_{PL_{rm}} \tag{6}$$

where β, γ, and η are three weight coefficients. By observation, we found that L^{*}_{Ad} is approximately ten times larger than L_{Ac}, thus we set γ to 0.1, both β and η to 1.0.

4.2 Metrics

For quantitative evaluations, we follow [2, 25] to adopt composition-relevant and graphic metrics to evaluate the performance of our model. Composition-relevant metrics include measuring text background complexity R_{com}, occlusion subject degree R_{shm}, and occlusion product degree R_{sub}. Graphic metrics consist.

of layout overlap R_{ove}, underlay overlap R_{und}, layout alignment R_{ali}, and the ratio of nonempty layouts R_{occ}. In addition, we introduce two metrics, R_{lac} and R_{plc}, to evaluate the model's performance on layout attribute and partial layout constraints, respectively. R_{lac} indicates the ratio of generated layouts that comply with the given attribute constraints. R_{plc} is used to quantify the average difference between given partial layout constraints and generated layouts. Combining the above metrics can reflect the model's performances regarding

graphic quality, product content relevance, layout attribute constraint, and partial layout consistency.

4.3 Layout attribute constraints

4.3.1 Comparison with image-aware methods

We first compare our method with image-aware methods: ContentGAN [50], CGL-GAN [2], and PDA-GAN [25]. Note that when the attribute is "layout with text but without any other class elements," IUC-Layout network does not generate underlays, rendering R_{und} irrelevant. Quantitative evaluations presented in Table 3 demonstrate that IUC-Layout network achieves the best performance in all other metrics except for R_{und}. A key strength of IUC-Layout network lies in its ability to generate image-aware layouts adhering to various attribute constraints.

Correspondingly, qualitative comparisons are presented in Fig. 3. Note that embellishments can overlap with any elements as their purpose is to enhance the layouts' aesthetics. Underlays are commonly used alongside texts to emphasize them. Columns 5 to 8 of Fig. 3 demonstrate that our model effectively aligns with and disentangles distinct layout attributes. For instance, in the 7th column, all generated layouts by IUC-Layout network include the logo (consistent), while excluding the embellishment (disentangled).

Interestingly, as shown in the 2nd, 3rd, and 4th columns in Fig. 3, previous works tend to create layouts with few or even no embellishment due to the low frequency (3.23%) of the embellishments in the CGL-Dataset. In contrast, our model consistently generates embellishments when the designated attribute is "layout with embellishment," as demonstrated in the 8th column. Moreover, the 4th and 5th rows in Fig. 3 highlight our model's superiority over CGL-GAN and PDA-GAN in terms of layout overlap and alignment.

4.3.2 Comparison with image-agnostic methods

We also compare IUC-Layout network with image-agnostic methods, including LayoutTransformer [22] and LayoutVTN [21]. Quantitative results in Table 4 demonstrate that our model outperforms LayoutTransformer and LayoutVTN in composition-relevant metrics (R_{com}, R_{shm}, and R_{sub}) across all attribute conditions. As shown in Fig. 4, layouts generated by our model are more effective in avoiding high-product-attention regions and human faces, enabling a comprehensive and visually pleasing presentation of product information.

4.3.3 Ablation studies for attribute losses

To validate the effectiveness of designed attribute losses, we conducted paired ablation studies for each attribute constraint. As shown in Table 5, after incorporating attribute

Table 5 Ablation studies on attribute losses. ✓ (×) denotes our model trained with (without) attribute losses L_A. Each pair of rows presents an experimental comparison under the same attribute constraint. Text as a sample indicates "layout with texts but without any other class elements"

Attribute	L_A	$R_{lac}\uparrow$	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
Text	×	**0.981**	35.08	11.68	0.724	0.0162	–	0.0122
Text	✓	0.973	**34.23**	**10.43**	**0.664**	**0.0129**	–	**0.0084**
Underlay	×	0.958	34.31	17.15	0.916	**0.0138**	0.8969	0.0048
Underlay	✓	**0.980**	**32.69**	**16.64**	**0.816**	0.0172	**0.9312**	**0.0030**
Logo	×	0.998	36.68	17.09	0.975	0.0529	0.9038	0.0157
Logo	✓	**1.000**	**35.93**	**16.27**	**0.936**	**0.0255**	**0.9226**	**0.0144**
Embellishment	×	0.989	35.25	**15.01**	0.916	**0.0261**	0.9035	**0.0079**
Embellishment	✓	**0.996**	**33.79**	15.66	**0.899**	0.0291	**0.9163**	0.0081
Unspecified	×	–	35.01	**15.19**	0.883	0.0321	0.8914	0.0092
Unspecified	✓	–	**33.06**	15.93	**0.826**	**0.0174**	**0.9221**	**0.0055**

The bold values indicate the best performance under the same constraints

Table 6 Quantitative evaluation on L_P and random mask PL_{rm}. *IcEI* means whether the method can handle partial layout constraints with in complete element information

Model	L_P	PL_{rm}	$R_{plc}\downarrow$	*IcEI*
CGL-GAN [2]			0.2895	×
PDA-GAN [25]			0.2693	×
CGL-GAN [2]	✓		0.0004	×
PDA-GAN [25]	✓		0.0004	×
CGL-GAN [2]	✓	✓	0.0006	✓
PDA-GAN [25]	✓	✓	0.0008	✓

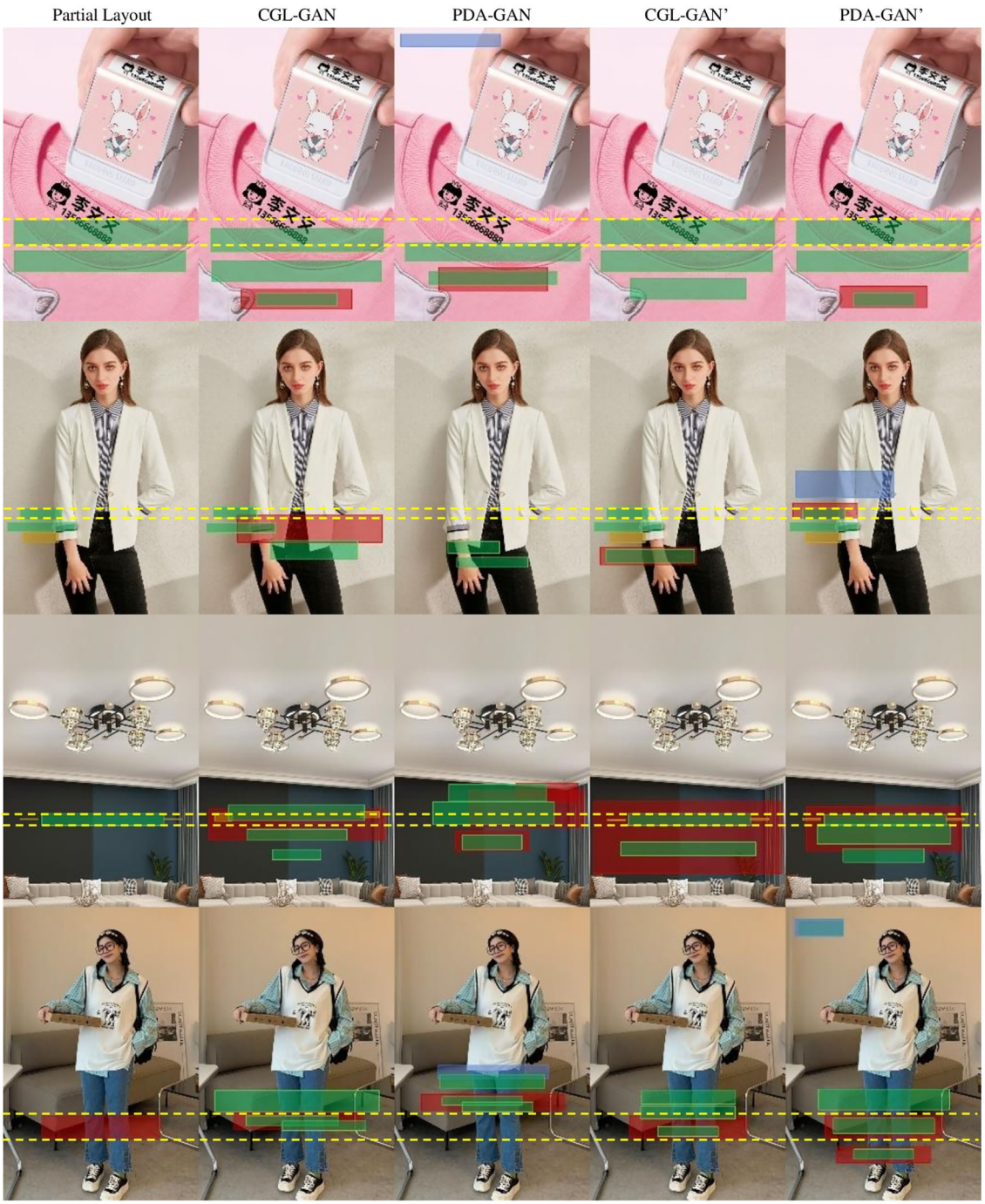


Fig. 5 Effects of L_P. The yellow dashed line is used to measure the alignment between the generated layouts and the given partial layout. CGL-GAN$^{'}$ and PDA-GAN$^{'}$ mean CGL-GAN and PDA-GAN with L_P, respectively

losses, the model exhibited a clear advantage in both composition-relevant and graphic metrics.

Additionally, we explored multiple ablation experiments on the attribute constraint module. One approach involved utilizing a linear layer to process a single value and connect it with feature maps, while another method replaced the Gaussian noise with the fixed mean. Unfortunately, neither of these methods yielded the anticipated results. Due to space constraints, we provide more comparative sample presentations in the supplementary material.

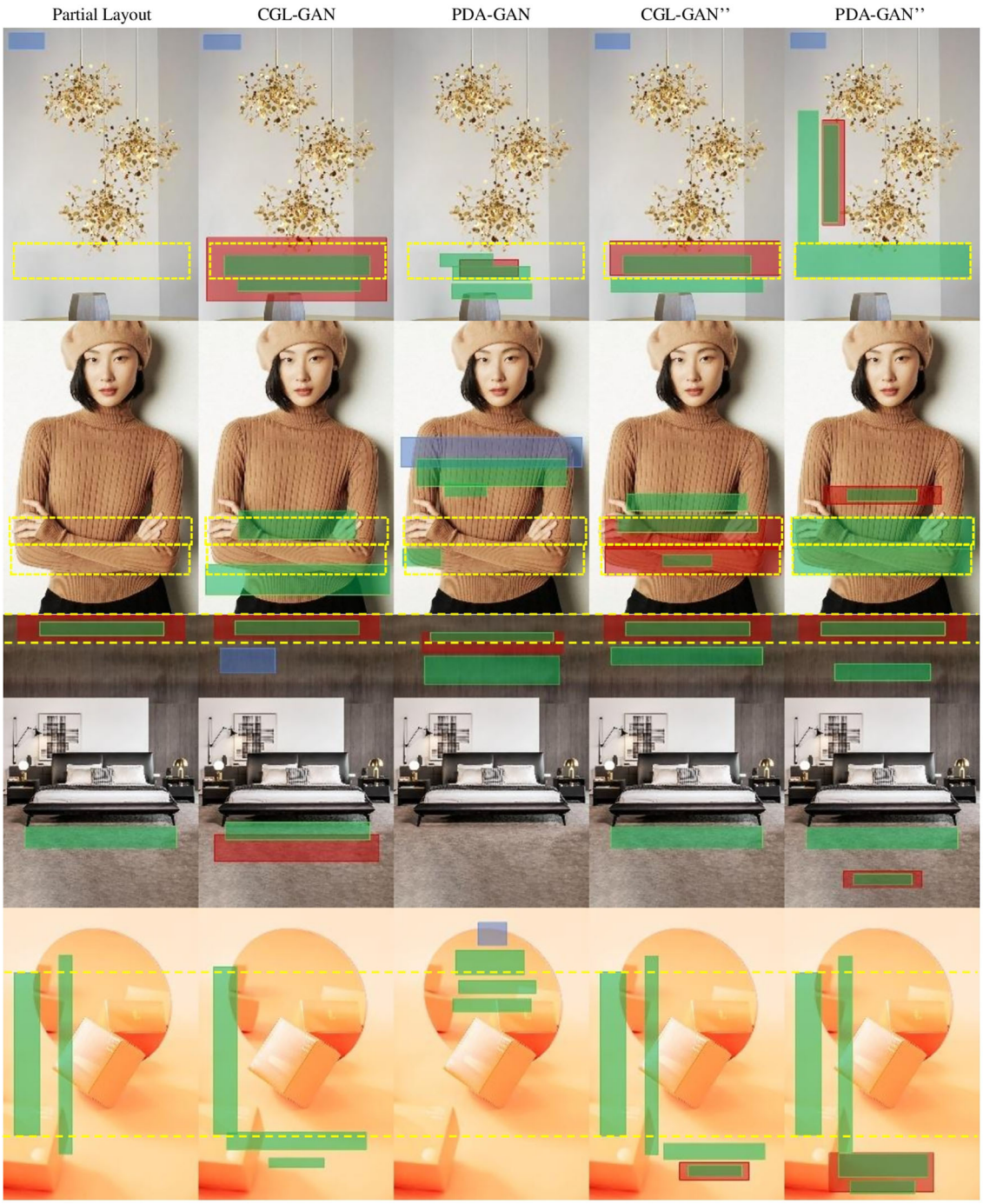


Fig. 6 Effects of $L_{PL_{rm}}$. The yellow boxes in the first two rows indicate the element with box coordinates but without class information. CGL-GAN" and PDA-GAN" mean CGL-GAN and PDA-GAN with $L_{PL_{rm}}$, respectively

4.4 Partial layout constraints

4.4.1 Effects of *Lp*

As shown in the first four rows of Table 6, introducing L_P to CGL-GAN (PDA-GAN) reduces the dissimilarity between generated layouts and provided information from 0.2895 (0.2693) to 0.0004 (0.0004). Layouts generated by CGL-GAN and PDA-GAN in the 1st and 3rd rows of Fig. 5 roughly match the given element's information but with distinct positional deviations. In contrast, models with L_P produce layouts that closely adhere to provided constraints. In the 2nd row, models without L_P generate layouts that even missing some elements from the partial layout. Notably, from the 4th and

Table 7 Cost comparison

Model	Parameters	FLOPs
CGL-GAN [2]	5.745×10^7	2.274×10^{11}
PDA-GAN [25]	3.806×10^7	1.929×10^{11}
IUC-Layout (Ours)	**3.773×10^7**	**1.528×10^{11}**

Bold numbers denote the best result

Table 8 User study

Model	$P_e \uparrow$	$P_b \uparrow$	$P_e^* \uparrow$	$P_b^* \uparrow$
CGL-GAN [2]	26.96	20.83	26.39	22.74
PDA-GAN [25]	26.55	23.13	26.03	21.07
IUC-Layout (Ours)	**46.49**	**56.04**	**47.58**	**56.19**

*denotes the professional group
Bold numbers denote the best result

5th columns, models with L_P generate layouts that not only maintain high consistency with the given partial layout but also precisely align newly generated element boxes with the provided element positions, effectively enhancing the layout aesthetics. Furthermore, in the 3rd and 4th rows, when the partial layout includes underlay (or text) element, the model can correspondingly generate text (or underlay), resulting in a harmonized layout.

4.4.2 Effects of PL_{rm}

As shown in Table 6, previous models without PL_{rm} cannot handle partial layout constraints with incomplete element information. The introduction of PL_{rm} slightly reduces performance compared to using L_P only, possibly due to the increased complexity of the task by PL_{rm}. In Fig. 6, the first partial layout includes a logo element and coordinates of another element without class information. The second sample provides the positions of two boxes without classes. The first two samples in Fig. 6 demonstrate that layouts generated by models with $L_{PL_{rm}}$ are consistent with complete elements and reasonably supplement incomplete elements. The last two rows show that models with $L_{PL_{rm}}$ also perform well in partial layout constraints with complete element information.

4.5 Computational complexity and user study

4.5.1 Computational complexity

As shown in Table 7, compared to other image-aware layout generation models, our model has the lowest number of parameters (3.773×10^7) and computational complexity (1.528×10^{11}). In testing, it only needs 4.1 ms to yield a layout on one NVIDIA V100 GPU. This signifies the suitability of our model for practical implementation.

4.5.2 User study

In addition to the general quantitative metrics, we also conducted a user study, as shown in Table 8, to accurately evaluate the model's performance. We randomly selected 60 test samples (20 with no user constraints, 20 with attribute constraints, and 20 with partial layout constraints). Each sample includes one product image and three corresponding predicted layouts (by CGL-GAN, PDA-GAN, and our model). We split participants into two groups (5 professional designers and 24 novice designers) and asked them to select eligible and best layouts from the three predicted layouts. The eligible-selected (best-selected) layout percentage P_e (P_b), which is the ratio of this model's vote count to the total vote count of all models, are shown in Table 8, revealing that our model's performance significantly outperformed other methods.

In addition to the aforementioned evaluations, further experimental details can be found in the supplementary.

5 Conclusions

Our proposed IUC-Layout network is designed for generating image-aware layouts with diverse user constraints. To generate layouts with specified attributes, we propose attribute-consistent and attribute-disentangled losses. We also introduce a random mask and partial layout loss to satisfy partial layout constraints, which can be easily applied to other methods. Quantitative and qualitative evaluations demonstrate that IUC-Layout network can generate high-quality image-aware layouts that adhere to various user constraints. In the future, we will further explore image-aware layout generation with intuitive user constraints, including but not limited to sequence constraints.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00371-024-03657-z.

Acknowledgements Weiwei Xu is partially supported by "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2023C01181). This paper is supported by Information Technology Center and State Key Lab of CAD&CG, Zhejiang University."

Author contributions C.X. (Chenchen Xu) and K.H. (Kaixin Han) developed the methodology and conducted the experiments. C.X. drafted the main manuscript text. K.H. and W.X. (Weiwei Xu) reviewed and edited the manuscript. W.X. supervised the project and provided guidance throughout the study. All authors read and approved the final manuscript.

Data availability No datasets were generated or analysed during the current study.

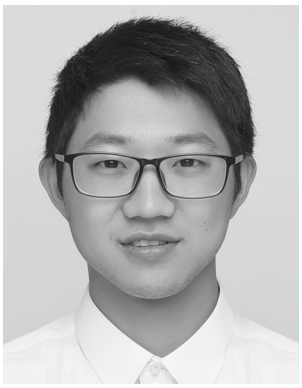
Declarations

Conflict of interest The authors declare no competing interests.

References

1. Li, J., Yang, J., Hertzmann, A., Zhang, J., Xu, T.: Layoutgan: Synthesizing graphic layouts with vector-wireframe adversarial networks. IEEE Trans. Pattern Anal. Mach. Intell. **43**(7), 2388–2399 (2021)
2. M. Zhou, C. Xu, Y. Ma, T. Ge, Y. Jiang, and W. Xu, (2022) "Composition-aware graphic layout GAN for visual-textual presentation designs," in IJCAI. ijcai. org. 4995–5001.
3. Kanungo, T., Mao, S.: Stochastic language models for style-directed layout analysis of document images. IEEE Trans. Image Process. **12**(5), 583–596 (2003)
4. E. Schrier, M. Dontcheva, C. Jacobs, G. Wade, and D. Salesin, (2008) "Adaptive layout for dynamically aggregated documents," in Proceedings of the 13th international conference on Intelligent user interfaces, 99–108
5. Hedjam, R., Nafchi, H.Z., Kalacska, M., Cheriet, M.: Influence of color-to-gray conversion on the performance of document image binarization: Toward a novel optimization problem. IEEE Trans. Image Process. **24**(11), 3637–3651 (2015)
6. X. Yang, T. Mei, Y. Xu, Y. Rui, and S. Li, (2016) "Automatic generation of visual-textual presentation layout." ACM Trans Multim. Comput Commun Appl. 12 2 1 22
7. S. Tabata, H. Yoshihara, H. Maeda, and K. Yokoyama,(2019) "Automatic layout generation for graphical design magazines," in SIGGRAPH Posters. ACM, 9:1–9:2
8. Y. Qiang, Y. Fu, Y. Guo, Z.-H. Zhou, and L. Sigal, (2016) "Learning to generate posters of scientific papers." in Proceedings of the AAAI Conference on Artificial Intelligence. https://doi.org/10.1609/aaai.v30i1.10000
9. Qiang, Y.-T., Fu, Y.-W., Yu, X., Guo, Y.-W., Zhou, Z.-H., Sigal, L.: Learning to generate posters of scientific papers by probabilistic graphical models. J. Comput. Sci. Technol. **34**, 155–169 (2019)
10. You, W.-T., Jiang, H., Yang, Z.-Y., Yang, C.-Y., Sun, L.-Y.: Automatic synthesis of advertising images according to a specified style. Frontiers of Information Technology & Electronic Engineering **21**(10), 1455–1466 (2020)
11. S. Guo, Z. Jin, F. Sun, J. Li, Z. Li, Y. Shi, and N. Cao, (2021) "Vinci: An intelligent graphic design system for generating advertising posters," in CHI. ACM, 577:1–577:17.
12. Pang, X., Cao, Y., Lau, R.W., Chan, A.B.: Directing user attention via visual flow on web designs. ACM Transactions on Graphics (TOG) **35**(6), 1–11 (2016)
13. Zhang, Y., Hu, K., Ren, P., Yang, C., Xu, W., Hua, X.-S.: Layout style modeling for automating banner design. Proceedings of the on Thematic Workshops of ACM Multimedia **2017**, 451–459 (2017)
14. S. Vempati, K. T. Malayil, V. Sruthi, and R. Sandeep, (2020) "Enabling hyper-personalisation: Automated ad creative generation and ranking for fashion e-commerce," in Fashion Recommender Systems. Springer, 25–48.
15. Liang, X., Lin, T.: Sketch2wireframe: an automatic framework for transforming hand-drawn sketches to digital wireframes in ui design. The Visual Comput. **40**, 1–11 (2023)
16. Calic, J., Gibson, D.P., Campbell, N.W.: Efficient layout of comic-like video summaries. IEEE Trans. Circuits Syst. Video Technol. **17**(7), 931–936 (2007)
17. Cohn, N.: Navigating comics: An empirical and theoretical approach to strategies of reading comic page layouts. Front. Psychol. **4**, 46474 (2013)
18. Wang, Z., Romat, H., Chevalier, F., Riche, N.H., Murray-Rust, D., Bach, B.: Interactive data comics. IEEE Trans. Visual Comput. Graphics **28**(1), 944–954 (2021)
19. Qiao, X., Cao, Y., Lau, R.W.: Design order guided visual note layout optimization. IEEE Trans. Visual Comput. Graphics **29**(09), 3922–3936 (2023)
20. A. A. Jyothi, T. Durand, J. He, L. Sigal, and G. Mori, (2019) "Layoutvae: Stochastic scene layout generation from a label set," in ICCV. IEEE, 9894–9903
21. D. M. Arroyo, J. Postels, and F. Tombari, (2021) "Variational transformer networks for layout generation," in CVPR. Computer Vision Foundation / IEEE, 13642–13652
22. K. Gupta, J. Lazarow, A. Achille, L. Davis, V. Mahadevan, and A. Shrivastava, (2021) "Layouttransformer: Layout generation and completion with self-attention," in ICCV. IEEE, 984–994
23. M. Hui, Z. Zhang, X. Zhang, W. Xie, Y. Wang, and Y. Lu, (2023) "Unifying layout generation with a decoupled diffusion model," CoRR, vol. abs/2303.05049
24. Y. Cao, Y. Ma, M. Zhou, C. Liu, H. Xie, T. Ge, and Y. Jiang, (2022) "Geometry aligned variational transformer for image-conditioned layout generation," in ACM Multimedia. ACM, 1561–1571
25. C. Xu, M. Zhou, T. Ge, Y. Jiang, and W. Xu, (2023) "Unsupervised domain adaption with pixel-level discriminator for image-aware layout generation," in CVPR. IEEE, 10114–10123
26. H. Hsu, X. He, Y. Peng, H. Kong, and Q. Zhang, (2023) "Posterlayout: A new benchmark and approach for content-aware visual-textual presentation layout," CoRR, vol. abs/2303.15937
27. F. Li, A. Liu, W. Feng, H. Zhu, Y. Li, Z. Zhang, J. Lv, X. Zhu, J. Shen, Z. Lin *et al.*, (2023) "Relation-aware diffusion model for controllable poster layout generation," in Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 1249–1258
28. D. Horita, N. Inoue, K. Kikuchi, K. Yamaguchi, and K. Aizawa, (2024) "Retrieval-augmented layout transformer for content-aware layout generation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 67–76
29. O'Donovan, P., Agarwala, A., Hertzmann, A.: Learning layouts for single-pagegraphic designs. IEEE Trans. Vis. Comput. Graph. **20**(8), 1200–1213 (2014)
30. Y. Xie, D. Huang, J. Wang, and C.-Y. Lin, (2021) "Canvasemb: Learning layout representation with large-scale pre-training for graphic design," in Proceedings of the 29th ACM international conference on multimedia, 4100–4108
31. N. Yu, C.-C. Chen, Z. Chen, R. Meng, G. Wu, P. Josel, J. C. Niebles, C. Xiong, and R. Xu, (2022) "Layoutdetr: detection transformer is a good multimodal layout designer," arXiv preprint arXiv:2212.09877
32. Xuan, Y., Song, C., Jin, J., Yang, B.: Cvae-layout: automatic furniture layout with constraints. The Visual Comput (2023). https://doi.org/10.1007/s00371-023-03204-2
33. Jacobs, C.E., Li, W., Schrier, E., Bargeron, D., Salesin, D.: Adaptive grid-based document layout. ACM Trans. Graph. **22**(3), 838–847 (2003)
34. R. Kumar, J. O. Talton, S. Ahmad, and S. R. Klemmer, (2011) "Bricolage: example-based retargeting for web design," in CHI. ACM, 2197–2206
35. Cao, Y., Chan, A.B., Lau, R.W.H.: Automatic stylistic manga layout. ACM Trans. Graph. **31**(1), 10 (2012)
36. P. O'Donovan, A. Agarwala, and A. Hertzmann, (2015) "Designscape: Design with interactive layout suggestions," in CHI. ACM, 1221–1224
37. H. Lee, L. Jiang, I. Essa, P. B. Le, H. Gong, M. Yang, and W. Yang, (2020) "Neural design network: Graphic layout generation with constraints," in ECCV (3), ser. Lecture Notes in Computer Science. Springer, 12348 491–506

38. C. Yang, W. Fan, F. Yang, and Y. F. Wang, (2021) "Layouttransformer: Scene layout generation with conceptual and spatial diversity," in CVPR. Computer Vision Foundation / IEEE, 3732–3741
39. M. Guo, D. Huang, and X. Xie, (2021) "The layout generation algorithm of graphic design based on transformer-cvae," *CoRR*, vol. abs/2110.06794
40. K. Kikuchi, E. Simo-Serra, M. Otani, and K. Yamaguchi, (2021) "Constrained graphic layout generation via latent optimization," in ACM Multimedia. ACM, 88–96
41. J. Li, J. Yang, J. Zhang, C. Liu, C. Wang, and T. Xu, (2021) "Attribute-conditioned layout GAN for automatic graphic design," IEEE Trans. Vis. Comput. Graph. 2710 4039–4048,. [Online]. Available: https://doi.org/10.1109/TVCG.2020.2999335
42. Z. Jiang, S. Sun, J. Zhu, J. Lou, and D. Zhang, (2022) "Coarse-to-fine generative modeling for graphic layouts," in AAAI. AAAI Press, 1096–1103
43. J. Zhang, J. Guo, S. Sun, J. Lou, and D. Zhang, (2023) "Layoutdiffusion: Improving graphic layout generation by discrete diffusion probabilistic models," CoRR, vol. abs/2303.11589
44. X. Kong, L. Jiang, H. Chang, H. Zhang, Y. Hao, H. Gong, and I. Essa, (2022) "Blt: Bidirectional layout transformer for controllable layout generation," in European Conference on Computer Vision. Springer, 474–490.
45. C. Cheng, F. Huang, G. Li, and Y. Li, "Play: Parametrically conditioned layout generation using latent diffusion," *CoRR*, vol. abs/2301.11529, 2023.
46. Z. Jiang, J. Guo, S. Sun, H. Deng, Z. Wu, V. Mijovic, Z. J. Yang, J.-G. Lou, and D. Zhang, (2023) "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, 18403–18412
47. N. Inoue, K. Kikuchi, E. Simo-Serra, M. Otani, and K. Yamaguchi, (2023) "Layoutdm: Discrete diffusion model for controllable layout generation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10167–10176
48. E. Levi, E. Brosh, M. Mykhailych, and M. Perez, (2023) "Dlt: Conditioned layout generation with joint discrete-continuous diffusion layout transformer," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2106–2115.
49. Fan, R., Wang, L., Liu, X., Im, S.K., Lam, C.T.: Real-scene-constrained virtual scene layout synthesis for mixed reality. The Visual Comput **40**(9), 6319–6339 (2023). https://doi.org/10.1007/s00371-023-03167-4
50. Zheng, X., Qiao, X., Cao, Y., Lau, R.W.H.: Content-aware generative modeling of graphic design layouts. ACM Trans. Graph. **38**(1), 15 (2019)
51. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, (2017) "Attention is all you need," in NIPS, 5998–6008
52. P. Zhang, C. Li, and C. Wang, (2020) "Smarttext: Learning to generate harmonious textual layout over natural image," in ICME. IEEE, 1–6
53. Li, C., Zhang, P., Wang, C.: Harmonious textual layout generation over natural images via deep aesthetics learning. IEEE Trans. Multim. **24**, 3416–3428 (2022)
54. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, (2020) "End-to-end object detection with transformers," in ECCV (1), ser. Lecture Notes in Computer Science, vol. 12346. Springer, 213–229
55. K. He, X. Zhang, S. Ren, and J. Sun, (2016) "Deep residual learning for image recognition," in CVPR. IEEE Computer Society, 770–778
56. T. Lin, P. Dollar, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, (2017) "Feature pyramid networks for object detection," in´CVPR. IEEE Computer Society, 936–944.
57. T. Karras, S. Laine, and T. Aila, (2019) "A style-based generator architecture for generative adversarial networks," in *CVPR*. Computer Vision Foundation / IEEE, 4401–4410
58. D. P. Kingma and J. Ba, (2015) "Adam: A method for stochastic optimization," in ICLR (Poster)
59. H. Chefer, S. Gur, and L. Wolf, (2021) "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in ICCV. IEEE. 387–396
60. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, (2021) "Learning transferable visual models from natural language supervision," in ICML, ser. Proceedings of Machine Learning Research, PMLR. 139 8748–8763



Chenchen Xu received the B.Sc., and M.sc. degree from Anhui Normal University, Wuhu, China, in 2016, and 2020, respectively. He is currently working toward the Ph.D. degree in the State Key Lab of CAD & CG, College of Computer Science and Technology, Zhejiang University, Hangzhou, China, under the supervision of Prof. W. Xu. His research interests include image processing and machine learning with a focus on deep learning, graphic layout generation and image matting.

Kaixin Han is a Ph.D. candidate attached to the College of Computer Science and Technology, Zhejiang University. With a background in human-computer interaction, his Ph.D. research focuses on computational aesthetics.

Weiwei Xu is a researcher with the State Key Lab of CAD & CG, College of Computer Science, Zhejiang University, awardee of NSFC Excellent Young Scholars Program in 2013. His main research interests include the digital geometry processing, physical simulation, computer vision, and virtual reality. He has published around 70 papers on international graphics journals and conferences, including 16 papers on ACM TOG. He is a member of the IEEE.