This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2025.3602846

GAN-based Domain Adaptation for Image-aware Layout Generation in Advertising Poster Design

Chenchen Xu, Min Zhou, Tiezheng Ge, and Weiwei Xu, *Member, IEEE*

***Abstract*—Layout plays a crucial role in graphic design and poster generation. Recently, the application of deep learning models for layout generation has gained significant attention. This paper focuses on using a GAN-based model conditioned on images to generate advertising poster graphic layouts, requiring a dataset of paired product images and layouts. To address this task, we introduce the Content-aware Graphic Layout Dataset (CGL-Dataset), consisting of 60,548 paired inpainted posters with annotations and 121,000 clean product images. The inpainting artifacts introduce a domain gap between the inpainted posters and clean images. To bridge this gap, we design two GAN-based models. The first model, CGL-GAN, uses Gaussian blur on the inpainted regions to generate layouts. The second model combines unsupervised domain adaptation by introducing a GAN with a pixel-level discriminator (PD), abbreviated as PDA-GAN, to generate image-aware layouts based on the visual texture of input images. The PD is connected to shallow-level feature maps and computes the GAN loss for each input-image pixel. Additionally, we propose three novel content-aware metrics to assess the model's ability to capture the intricate relationships between graphic elements and image content. Quantitative and qualitative evaluations demonstrate that PDA-GAN achieves state-of-the-art performance and generates high-quality image-aware layouts.**

***Index Terms*—Graphic layout, image-aware layout generation, domain gap, domain adaptation, advertising poster design.**

I. Introduction

GRAPHIC layout is essential for the design of posters, magazines, comics, and webpages. Recently, generative adversarial networks (GANs) have been applied to synthesize graphic layouts by modeling the geometric relationships among different types of 2D elements, such as text and logo bounding boxes [1], [2]. Fine-grained control over the layout generation process can be achieved using conditional GANs. The conditions may include image content and attributes of graphic elements, such as category, area, and aspect ratio [3]. Especially, image content plays an important role in generating image-aware graphic layouts of posters and magazines [4].

This paper focuses on the GAN-based image-aware graphic layout generation for advertising poster design, where the layout involves arranging various elements. As shown in Fig. 1, our graphic layout generation task involves arranging four classes of elements: logos, text, underlays, and embellishment elements, at appropriate positions based on the product image.

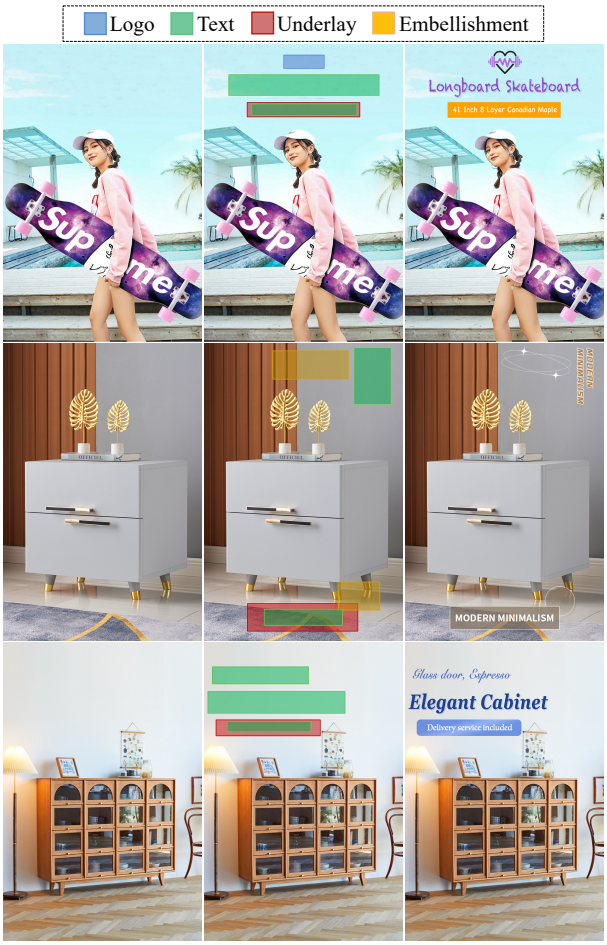


Fig. 1: **Examples of image-aware graphic layout generation for advertising posters.** Our model generates graphic layouts (middle) with multiple elements conditioned on product images (left). Designers or even automatic rendering programs can use these layouts to render advertising posters (right).

The core challenge lies in modeling the relationship between the image content and the layout elements, enabling the neural network to learn how to produce aesthetically pleasing arrangements around the product image. This task can be formulated as a direct set prediction problem, as described in [5].

Constructing a high-quality layout dataset for training image-aware graphic layout generation models is labor-intensive, as it requires professional stylists to design the arrangement of elements to create paired product images and layout data items. To reduce the workload, we collect designed

Chenchen Xu is with Anhui Normal University, Wuhu, China; Zhejiang University, Hangzhou, China; and Shenzhen Bay Laboratory, Shenzhen, China (e-mail: xuchenchen@zju.edu.cn).

Min Zhou and Tiezheng Ge are with Alibaba Group, Hangzhou, China (e-mail: yunqi.zm@alibaba-inc.com; tiezheng.gtz@alibaba-inc.com).

Weiwei Xu is with Zhejiang University, Hangzhou, China (e-mail: xww@cad.zju.edu.cn).

Corresponding author: Weiwei Xu.

poster images and product images to construct a large content-aware graphic layout dataset (CGL-Dataset), which includes 60,548 advertising posters with annotated layout information and 121,000 clean product images without annotations. The graphic elements imposed on the posters are removed through inpainting [6] and annotated with their geometric arrangements, resulting in the state-of-the-art CGL-Dataset. While the CGL-Dataset is substantially beneficial to the training of image-aware networks, the inpainting artifact introduces a domain gap between inpainted posters (source domain data) and clean product images (target domain data).

We propose two approaches to narrow this domain gap. The first approach, CGL-GAN, applies a simple yet effective blurring operation to reduce the distinction between inpainted and non-inpainted regions. This operation smooths both areas simultaneously, narrowing the domain gap and facilitating the generation of image-aware graphic layouts. Although effective, blurred images may lose the delicate color and texture details of products, potentially resulting in undesirable occlusion or placement of graphic elements. The second approach employs an unsupervised domain adaptation technique [7]–[11] to bridge the domain gap between clean product images and inpainted posters in the CGL-Dataset, significantly improving the quality of generated graphic layouts. The goal of this approach is to align the feature spaces of inpainted posters and clean product images, i.e., the source and target domains. To achieve this, we propose a GAN with a pixel-level domain adaptation discriminator, called PDA-GAN, which allows for fine-grained control over feature space alignment. It is inspired by PatchGAN [12], but it does not directly adapt to pixel-level alignment in our task.

The pixel-level discriminator (PD) designed for domain adaptation can replace the Gaussian blurring step, enabling the network to better model the visual and texture details of the product image. The PD is connected to shallow-level feature maps, as the inpainted regions are typically small relative to the whole image and may be difficult to detect at deeper levels with large receptive fields. Additionally, the PD is composed of only three convolutional layers, and its number of parameters is less than 2% of those in the CGL-GAN discriminator. This design significantly reduces the memory and computational cost introduced by the PD. A total of 120,000 target domain images were collected to support the training of PDA-GAN.

Existing metrics [13]–[15] consider only the relationships among graphic elements and ignore the relationship between graphic elements and image content. Therefore, we propose three novel content-aware metrics to evaluate our methods in terms of image relevance. Considering the particularity and complexity of the image-aware graphic layout generation task, we redesign three content-agnostic graphic metrics. In addition to the conventional Fréchet Inception Distance (FID) [16], [17], we further introduce a content-aware variant, termed cFID, as an additional evaluation metric. Experimental results demonstrate that PDA-GAN achieves state-of-the-art (SOTA) performance. It outperforms CGL-GAN (ContentGAN [4]) on the CGL-Dataset and achieves relative improvements of 6.21% (26.41%), 17.5% (25.23%), 14.5% (39.81%), and 19.07% (52.89%), on the background complexity, subject occlusion degree, product occlusion degree, and cFID metrics, respectively, leading to significantly improved visual quality of generated graphic layouts in many cases.

This paper is an extension of our previous work presented in [18] and [19]. Building upon that foundation, we further present a detailed analysis of the CGL-Dataset, a formalized introduction of evaluation metrics, and additional experimental results on domain adaptation. In summary, the main contributions of this work are as follows:

- We contribute a large-scale layout dataset that covers a wide variety of promotional products and professionally designed layouts. To the best of our knowledge, this is the first large-scale dataset focused on advertising poster layout design.
- We propose PDA-GAN, which incorporates a novel pixel-level discriminator that operates on shallow-level feature maps to bridge the domain gap between clean product images and annotated inpainted posters in the CGL-Dataset.
- To evaluate the global and spatial information of input images learned by the model, we propose three novel content-aware metrics. Given the particularity and complexity of advertising poster graphic design, we redesign three content-agnostic graphic metrics, along with a content-aware version of FID.

More importantly, both quantitative and qualitative evaluations demonstrate that our method achieves SOTA performance and can generate high-quality image-aware graphic layouts for advertising posters.

II. Related Work

A. Image-agnostic Layout Generation

Early works [20]–[25] often utilize templates and heuristic rules to generate layouts. These approaches rely on professional knowledge and often fail to produce flexible and diverse layouts limited to their hand-crafted rules. LayoutGAN [2] is the first method to apply generative networks (in particular, GANs) to synthesize layouts and use self-attention to model element relationships. LayoutVAE [14] and LayoutVTN [15] both follow and apply VAE and autoregressive methods. Diffusion-based models such as LayoutDM [26] and Layout-Diffusion [27] have also been proposed for layout generation. Meanwhile, some conditional methods have been proposed to guide the layout generation process [3], [28]–[31]. The constraints are in various forms, such as scene graphs, element attributes, and partial layouts. However, in general, these methods mainly focus on modeling the internal relationship between graphic elements and rarely consider the relationship between layouts and images.

B. Image-aware Layout Generation

In recent years, various image-aware layout generation methods have emerged, including LLM-based [32] approaches, reflecting the growing interest and rapid progress in this field. In the generation of magazine page layouts, ContentGAN [4] was the first to model the relationship not only between layout

TABLE I: **Different dataset of poster layout.** The symbol $\mathcal{S}$ ($\mathcal{T}$) represents the source (target) domain data.

Dataset	Status	Train $\mathcal{S}$	Test $\mathcal{S}$	Train $\mathcal{T}$	Test $\mathcal{T}$	Element classes	Content	Product category
NDN [28]	Private	0	500	0	0	Text, logo, image, button	Empty	Car
ICVT [44]	Private	105,862	11,762	0	166	Text, logo, underlay	Product	Clothing, electronics, cosmetics, etc.
PKU [33]	Public	9,974	0	0	905	Text, logo, underlay	Product	Clothing, electronics, cosmetics, etc.
CGL(Ours)	Public	54,546	6,002	120,000	1,000	Text, logo, underlay, embellishment	Product	Clothing, electronics, cosmetics, etc.

elements but also between layouts and images. However, high-quality training data is relatively scarce, as it requires professional stylists to manually design layouts in order to obtain paired clean images and layout annotations. To address this, ContentGAN uses white patches to mask the graphic elements on magazine pages and employs these processed pages as substitutes for clean images during training. In the context of poster layout generation, we tackle the same issue by applying image inpainting [6], [33] to remove graphic elements from posters, followed by a Gaussian blur applied to the entire image to mitigate inpainting artifacts. While this blur strategy effectively narrows the domain gap between inpainted and clean images, it may also degrade fine color and texture details, leading to suboptimal occlusion or element placement. In this paper, we demonstrate that a pixel-level discriminator designed for domain adaptation can achieve similar goals while avoiding the negative side effects of blurring.

C. Unsupervised Domain Adaptation

Unsupervised domain adaptation [10] aims at aligning the disparity between domains such that a model trained on the source domain with labels can be generalized into the target domain, which lacks labels. Many related methods [10], [11], [34]–[42] have been applied for object recognition, detection and segmentation. Among these methods, [10], [38], [39], [41] leverage adversarial domain adaptation approach [43]. A domain discriminator is employed and outputs a probability value indicating the domain of the input data. In this way, the generator can extract domain-invariant features and bridge the semantic or stylistic gap between the two domains. However, it does not work well when applied directly to our problem, since the inpainted area is small compared to the whole image and is difficult to discern at deep levels. Therefore, we design a pixel-level discriminator, which is connected to shallow-level feature maps and computes GAN loss for each input-image pixel, to effectively solve this.

III. Dataset and Representation

A. Different Dataset of Poster Layout

Existing publicly available datasets [45]–[48] predominantly focus on the relationships between graphic elements, without taking into account the content of the background image. These datasets are inadequate for the task of generating poster layouts. In recent years, a limited number of datasets related to poster layout have been presented, as shown in Tab. I. Since some of these datasets are still not publicly available to this day, we have made every effort to gather information from their source papers. NDN [28] presented a banner layout dataset consisting of 500 car advertising posters, designed to validate content-agnostic layout generation methods. This dataset lacks a substantial number of training samples and exhibits a limited variety of poster categories. ICVT [44] offers a large-scale dataset of 117,624 poster layouts, to some extent, facilitating the development of image-aware approaches. Unfortunately, this dataset is not publicly accessible and includes only 166 target domain images for testing, which is insufficient to effectively validate model performance. PKU [33] released a relatively small poster layout dataset, comprising 9,974 annotated source domain images and 905 target domain images. Similar to ICVT, the layout elements are classified into types of text, logo, and underlay, excluding embellishments, resulting in a lack of diversity.

To this end, we provide and release a substantial dataset named CGL-Dataset, which consists of 60,548 (54,546 for training, 6,002 for test) inpainted posters with annotated layout information and 121,000 (120,000 for training, 1,000 for test) clean product images. It shows advantages in multiple classes of elements (e.g., text, logo, underlay, and embellishment), a variety of products (e.g., clothing, electronics, cosmetics, etc., as shown in the upper right of Fig. 2), and diversity of layout positions (e.g., top-left, top-right, bottom, etc., as shown in the bottom right of Fig. 2). From the left part of Fig. 2, each sample in the source domain comprises five components: poster $\boldsymbol{x}_{pst}$, inpainted poster $\boldsymbol{x}_{pst}^{inp}$, salient map $\boldsymbol{x}_{pst}^{sal}$, layout annotations $\boldsymbol{l}_{GT}$, and white-patch map $\boldsymbol{x}_{pst}^{wp}$. The posters were collected with formal authorization from approved e-commerce platforms, ensuring that all data acquisition complies with copyright regulations. Layout annotations were subsequently performed on these posters in a systematic manner by trained personnel, following consistent labeling standards. The annotation information forms the graphic layout, which consists of n variable-length elements, represented as $\{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n\}$. Each element $\boldsymbol{e}$ is represented with its type c and bounding box $b = [x, y, w, h]$. (x, y) represents the top-left coordinates, and w (h) represents the width (height) of the bounding box. After obtaining the annotations, we utilized the pretrained InpNet [6] to perform inpainting on the annotated element box areas, resulting in $\boldsymbol{x}_{pst}^{inp}$. Following that, we used pretrained SalNet [49] to extract the salient map $\boldsymbol{x}_{pst}^{sal}$ from $\boldsymbol{x}_{pst}^{inp}$. Finally, we represent the annotation information in a binary image, values of pixels in element boxes region set to 1 and 0 elsewhere, dubbed white-patch map $\boldsymbol{x}_{pst}^{wp}$.

It is worth noting that in the task of image-aware layout generation, we are the first to explicitly recognize and address the domain gap. Consequently, our dataset is the only one that provides a large amount of target domain training data, including clean product images $\boldsymbol{x}_{img}$ and corresponding saliency maps $\boldsymbol{x}_{img}^{sal}$. Although background images without any graphic elements may offer a more fundamental basis for layout generation, obtaining such data at scale remains a con-

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2025.3602846

Fig. 2: **Examples of CGL-Dataset.** The left part represents the six components of information contained in each sample of the dataset. In the upper right corner are examples of different product types, and in the lower right corner are examples with different layout positions.

siderable challenge. This is primarily because clean product images typically require manual design and post-processing by professional designers, making large-scale collection highly labor-intensive and costly. In this work, we instead choose to annotate directly on collected posters, aiming to balance the trade-off between data quality and dataset construction feasibility. Nevertheless, exploring layout generation from clean background images remains a compelling direction for future research.

B. Domain Gap Visualization

Different marginal distributions across domains are referred to as domain gap [10]. In this work, the paired images and layouts in the existing dataset [18] are collected by inpainting [6] and annotating posters, respectively. A domain gap exists between inpainted posters (source domain data) and clean product images (target domain data).

To illustrate the domain gap, Fig. 3 shows examples of source and target domain images. We selected three clean product images from the target domain, added graphic elements to create posters, and then applied inpainting to these posters to generate the source domain data. This process resulted in distorted and blurred inpainted regions, which constitute a pixel-level domain gap.

IV. Our Model

This paper focuses on bridging the domain gap while preserving fine-grained details, such as color and texture, to generate image-aware graphic layouts for poster design. To achieve this, we introduce two GAN-based models, namely CGL-GAN and PDA-GAN, which share a common generator but employ distinct discriminators. CGL-GAN uses Gaussian-blurred inpainted posters as input to reduce the domain gap. Its discriminator adopts a structure similar to that of [5], judging whether the input image–layout pairs are real (annotated layout) or fake (generated layout), and outputs a probability score. Although effective in narrowing the domain gap, blurred images may lose the fine color and texture details of products, leading to unpleasant placement or occlusion of graphic elements.

Therefore, the second approach combines unsupervised domain adaptation techniques to design a GAN with a novel pixel-level discriminator (PD), called PDA-GAN, to generate graphic layouts according to image contents. As shown in Fig. 4, our network mainly has two sub-networks: the layout generator network that takes the image ($\boldsymbol{x}_{pst}^{inp}$ or $\boldsymbol{x}_{img}$) and its salient map ($\boldsymbol{x}_{pst}^{sal}$ or $\boldsymbol{x}_{img}^{sal}$) as the input to generate graphic layout and the convolutional neural network for pixel-level discriminator. In this section, we will describe the details of the pixel-level discriminator and the layout generator network, respectively.

A. Pixel-level Discriminator

The design of the pixel-level discriminator is based on the observation that the domain gap between inpainted images $\boldsymbol{x}_{pst}^{inp}$ and clean product images $\boldsymbol{x}_{pst}$ primarily exists at the pixels synthesized during the inpainting process. Therefore, during the discriminator or generator pass in Fig. 4, both inpainted and clean images are fed into the discriminator. When

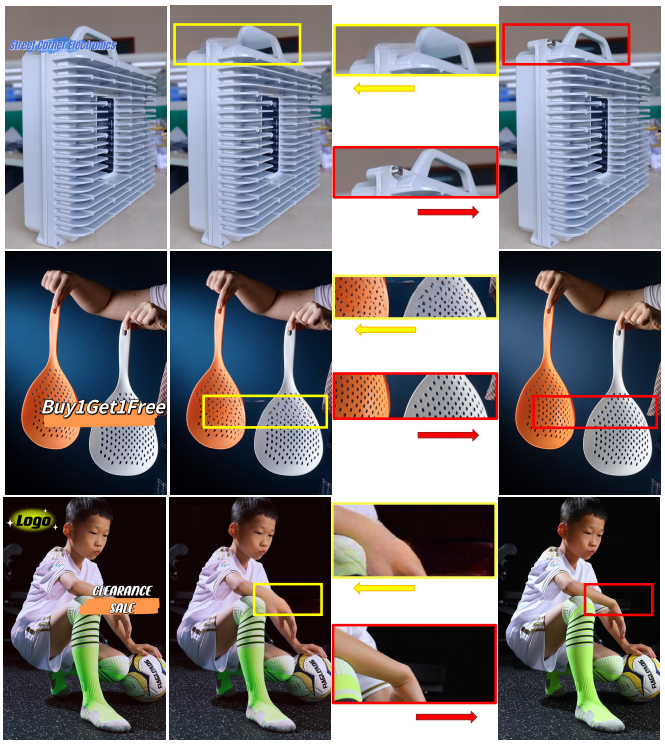


Fig. 3: **Domain gap visualization.** To illustrate the domain gap, we manually design several posters based on clean product images (target data), and then inpaint the graphic elements to generate the corresponding source data. The visual content in the inpainted areas (highlighted with yellow boxes) appears distorted and blurred compared to the original content (highlighted with red boxes).

updating the discriminator, we encourage the discriminator to detect the inpainted pixels for $\boldsymbol{x}_{pst}^{inp}$ in the source domain. In contrast, when updating the generator, we leverage the pixel-level discriminator to encourage the generator to output shallow feature maps that can fool the discriminator, which means that, even for the feature map computed for $\boldsymbol{x}_{pst}^{inp}$, the discriminator's ability to detect inpainted pixels should be weakened fast. In this way, when training converges, the feature space of source and target domain images should be aligned.

Our pixel-level discriminator network consists of three transposed convolutional layers with filter size 3×3 and stride 2. Its input is the feature map from the first residual block in the multi-scale CNN. These transposed convolutional layers upsample the feature map, and the final output can be resized to exactly match the dimensions of the input image, facilitating the computation of the discriminator's training loss.

To calculate the loss L_{PD} for each pixel in the input image, we use the white-patch map to identify whether a pixel has been inpainted. In this map, a pixel value is set to 1 if the corresponding pixel in the input image has been processed by inpainting; otherwise, it is set to 0. For clean images in the target domain, all pixel values in the white-patch map are 0.

When updating the discriminator in the GAN training, the pixel-level discriminator takes shallow-level feature maps as input and outputs a map with one channel whose dimension is consistent with the input image. The loss L_{PD} used to train the discriminator is a mean absolute error (MAE) loss or L1 norm between the white-patch map of input images and the output map. It is defined as:

$$L_{PD} = \frac{1}{N_p} \sum_{i=1}^{N_p} (|\mathbf{P}_i^{s,w} - \mathbf{P}_i^{s,o}| * \alpha + \left|\mathbf{P}_i^{t,w} - \mathbf{P}_i^{t,o}\right| * \beta), \tag{1}$$

where N_p denotes the number of white-patch map pixels, and $\mathbf{P}_i$ indicates the predicted or ground-truth map for the i_{th} image. The superscripts of $\mathbf{P}_i$ specify the domain and type: s for source or t for target, and o for prediction or w for ground truth. The two coefficients, α and β, are employed to strike a balance between the white-patch maps of the source and target domains. Since the area of the inpainted pixels in the white-patch map is usually small, we set the value of α to 2 and β to 1.

We utilize one-side label smoothing [50], [51] to improve the generalization ability of the trained model. Since the inpainted areas occupy a small proportion of the input image, we only apply label smoothing for pixels not in the inpainted area (i.e., pixels with a value of 0 in the white-patch map), denoted as one-target label smoothing in our experiments. Precisely, we only set 0 to 0.2 in the ground truth white-patch map.

B. Layout Generator

The architecture design of the layout generator network follows the principle of DETR [5], which has three modules: a multi-scale convolutional neural network (CNN) used to extract image features [52], [53], a transformer encoder-decoder that takes layout element queries as input to model the relationships between layout elements and the product image [54], and two fully connected layers to predict the element class and its bounding box based on the element features output by the transformer decoder.

Concatenated $\boldsymbol{x}_{pst}^{inp}$ with $\boldsymbol{x}_{pst}^{sal}$ (or $\boldsymbol{x}_{img}$ with $\boldsymbol{x}_{img}^{sal}$) is fed into the multi-scale CNN, a ResNet50 [52] whose input channels are changed to four and the part after its final convolutional layer is removed. For the reason that image visual-texture content not only means deep-level semantics such as subject locations, but also includes shallow-level features like region complexity, we introduce a multi-scale strategy on the last two convolutional blocks following FPN [55]. There is a slight difference that we do not generate layouts on each scale separately, like detection networks often do, but concatenate the fused and upsampled features as one. We denote $\boldsymbol{F}_j$ the feature maps of the j-th convolutional block. Then the multi-scale features can be computed as:

$$\begin{aligned} &\boldsymbol{F}_j^{'} = \mathbf{Conv_{11}}(\boldsymbol{F}_j)\ ; \quad \boldsymbol{F}_j^{up} = \mathbf{Upsample}(\boldsymbol{F}_j^{'})\ ; \\ &\boldsymbol{F}_j^{fused} = \mathbf{Concat}(\boldsymbol{F}_j^{up}, \mathbf{Conv_{33}}(\boldsymbol{F}_j^{up} + \boldsymbol{F}_{j-1}^{'})) \end{aligned} \tag{2}$$

where $\mathbf{Conv_{11}}$, $\mathbf{Conv_{33}}$, $\mathbf{Upsample}$ and $\mathbf{Concat}$ are network operations of convolution, up-sampling, and concatenation respectively. Although higher-resolution features can

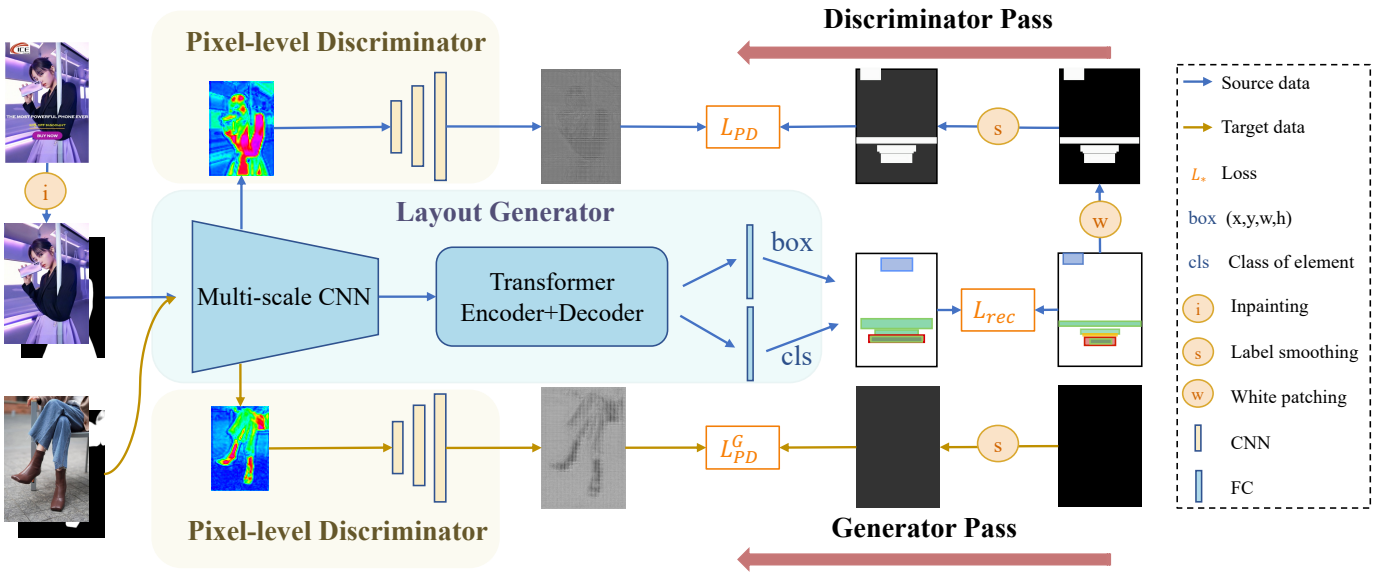


Fig. 4: **The architecture of our network.** Annotated posters (source domain data) must be inpainted before input to the model. The model has both reconstruction and GAN loss when training with source domain data, while only a GAN loss is used when training with target domain data. Please refer to Sec. IV for the definition of each loss term: L_{PD}, L^G_{PD}, and L_{rec}. During the discriminator or generator pass, both inpainted and clean images are fed into the discriminator.

provide more details, we empirically find fusing features from the last two blocks yields good results and is more efficient for training.

The multi-scale features are projected into d channels and then flattened by channel as the input to the transformer encoder. The encoder uses a standard transformer architecture to further refine the image features. The decoder utilizes cross-attention to learn the relationship between the image content and the graphic layouts. Both the encoder and decoder have six layers, and the hidden dims are d. Position encodings are added in both the encoder and decoder. Finally, the decoder features of each element are passed through two fully connected layers (FCs) to predict the corresponding class $\boldsymbol{c}$ and box coordinates $\boldsymbol{b}$ logits. The final class and box results are computed using the softmax and sigmoid functions.

When updating the generator network in the GAN training, we aim to fool the updated discriminator in the detection of inpainted pixels. Therefore, the loss L_{PD} is modified to penalize the generator network if the discriminator outputs a pixel value of 1. Thus, we have:

$$L^G_{PD} = \frac{1}{N_p}\sum_{i=1}^{N_p}\Big(\Big|\hat{\mathbf{P}}^s_i - \mathbf{P}^{s,o}_i\Big| * \alpha + \Big|\mathbf{P}^{t,w}_i - \mathbf{P}^{t,o}_i\Big| * \beta\Big), \tag{3}$$

where the values of pixels in $\hat{\mathbf{P}}^s_i$ are all set to 0.2. The training loss for the layout generator network is as follows:

$$L_G = L_{rec} + \gamma * L^G_{PD}, \tag{4}$$

where the value of the weight coefficient γ is set to 6, and the L_{rec} is the reconstruction loss to penalize the deviation between the graphic layout generated by the network and the annotated ground-truth layout for the inpainted images in the source domain. We calculate the reconstruction loss L_{rec} as the direct set prediction loss in [5].

V. Metrics

To better evaluate the performance of poster layout generation, we propose three novel content-aware metrics: R_{com}, R_{shm}, and R_{sub}, which assess background complexity, subject occlusion, and product occlusion, respectively. These content-aware metrics are particularly important in the context of this research. In addition, we introduce cFID, a content-aware variant of the standard FID [16], [17], to more effectively evaluate the semantic alignment between layout structures and image content. We also introduce four conventional graphic metrics: R_{ove}, R_{und}, R_{ali}, which quantify layout overlap, underlay overlap, and element alignment, respectively, along with FID [17], [28], which evaluates the overall distributional similarity between generated and real layouts. Moreover, we include the metric R_{occ} to measure the ratio of non-empty layouts predicted by the models. In this section, we will provide formal definitions of these metrics and use them to demonstrate the effectiveness of our approach.

A. Content-aware Metrics

Since there are no existing metrics specifically designed for the task of image-aware layout generation, we propose three novel content-aware metrics and a redesigned version of FID, termed cFID, tailored for image-aware layout evaluation. The metric R_{com} is designed to evaluate the visual complexity of the background in the regions where text elements (without underlays) are placed, which may affect text readability. A visually cluttered background—such as one with strong edges, textures, or noise—can reduce the clarity of overlaid text and impair the viewer's ability to quickly perceive the text

information. To quantify such background complexity, we compute the average gradient magnitude within the text-only bounding boxes. Specifically, for each pixel, the gradients along the x and y directions are computed using the Sobel operator, and the magnitude (i.e., the Euclidean norm) of the gradient vector is calculated. The final score R_{com} is defined as the average gradient magnitude over all pixels within the predicted text regions. A higher R_{com} indicates more complex background textures and thus potentially lower text readability. The metric R_{com} is computed as:

$$R_{com} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{|R_i|}\sum_{p\in R_i}\|\nabla \boldsymbol{x}_p\|_2\right) \tag{5}$$

where N is the number of text-only elements, and $|R_i|$ denotes the number of pixels within the i-th text region. $\nabla \boldsymbol{x}_p$ denotes the Sobel gradient vector at pixel p, and $\|\nabla \boldsymbol{x}_p\|_2$ represents its ℓ_2-norm, i.e., the magnitude of the gradient.

The metric R_{shm} is proposed to evaluate how severely graphic elements occlude the primary subject or product within the background image. A key aesthetic principle in advertising poster design is the clear visibility of core visual content (e.g., people or products), which should remain visually salient after the layout is applied. Therefore, a lower R_{shm} indicates less visual interference from layout elements and better preservation of the original salient regions. To quantify the semantic impact of layout occlusion, we compute the perceptual difference between the saliency map before and after layout masking. Specifically, we extract high-level semantic representations from a pretrained VGG16 [56] network by feeding it two saliency-based images: one is the original salient map x^{sal}, and the other is the same map with layout elements masked out, denoted as (x^{sal}, y^l). The Euclidean distance between the corresponding output logits from VGG16 reflects how much the layout occludes semantically important content in the image:

$$R_{shm} = L_2[\boldsymbol{VGG}(x^{sal}), \boldsymbol{VGG}(x^{sal}, y^l)]. \tag{6}$$

Here, x^{sal} represents the input saliency map, and y^l denotes the binary mask of the predicted layout. This distance reflects how much the layout disrupts the semantic content captured by the saliency map.

To compute R_{sub}, we aim to evaluate how much the layout design interferes with the recognizability of products. Unlike general saliency maps that highlight visually prominent areas, we leverage CLIP-based attention maps conditioned on product category tags, which capture semantic relevance between image regions and textual queries. Specifically, we extract product category names from product pages and use them as text prompts to query attention maps from a pretrained CLIP model[1] [57], [58]. The attention values within each layout bounding box are summed to reflect potential occlusions over semantically important regions.

$$R^s_{sub} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{Rgn}(\boldsymbol{CLIP}(Map^s)), \tag{7}$$

[1]https://github.com/hila-chefer/Transformer-MM-Explainability

where R^s_{sub} denotes the R_{sub} value for sample s, Map^s is the CLIP attention map for s, and $\boldsymbol{Rgn}$ extracts attention values within the predicted layout regions. A lower R_{sub} score suggests that layout elements better preserve semantically critical product areas. Compared to general saliency, CLIP attention maps are used here because they better reflect concept-level localization and can handle diverse products that may not be visually salient but are semantically important.

Fréchet Inception Distance (FID) [16], [17] is widely used to evaluate the similarity between two image distributions based on deep features. However, since our task focuses on image-aware layout generation and the test set lacks ground-truth layout annotations, the standard FID between predicted and real layouts cannot be directly applied. To address this, we propose a redesigned version of FID, termed cFID, to indirectly assess layout–image harmony. Specifically, we first use an InpNet [6] to erase the regions occupied by predicted layout elements on test images. Then, we compute the cFID value between the original image set $\boldsymbol{x}_{img}$ and the inpainted image set $\boldsymbol{x}^{inp}_{img}$ to evaluate the semantic disruption caused by the layouts. A lower value of this image-aware layout cFID indicates better preservation of image semantics and better coordination between layout and image content. The cFID is computed as follows:

$$\begin{aligned}\text{cFID} = &\left\|\mu_{\boldsymbol{x}_{img}} - \mu_{\boldsymbol{x}^{inp}_{img}}\right\|_2^2 \\ &+ \text{Tr}\big(\Sigma_{\boldsymbol{x}_{img}} + \Sigma_{\boldsymbol{x}^{inp}_{img}} - 2\cdot(\Sigma_{\boldsymbol{x}_{img}}\Sigma_{\boldsymbol{x}^{inp}_{img}})^{\frac{1}{2}}\big)\end{aligned} \tag{8}$$

where $\mu_{\boldsymbol{x}_{img}}$ and $\Sigma_{\boldsymbol{x}_{img}}$ are the mean and covariance of the Inception features for the original images, and $\mu_{\boldsymbol{x}^{inp}_{img}}$ and $\Sigma_{\boldsymbol{x}^{inp}_{img}}$ are the mean and covariance for the inpainted images.

B. Graphic Metrics

Content-agnostic graphic metrics, such as overlap and alignment of layout elements, are described in [3], [59]. However, these metrics overlook the different element types and relationships between different types of elements in the image-aware layout generation task. As mentioned in the paper, the graphic layouts of advertising posters encompass four types of elements: logos, texts, underlays, and embellishments. Specifically, the underlay elements are allowed to overlap with any other elements to improve the readability of texts, since it is possible that the desirable color of text is not salient on a background image. The layout also allows an embellishment to overlap with other elements, except for other embellishments, as embellishments are typically used to decorate posters to enhance the aesthetics.

We follow the equation in [3] to calculate the layout overlap metric as follows:

$$R^{\mathbf{e}}_{ove} = \sum_{i\in\mathbf{e}}\sum_{(j\in\mathbf{e})\neq i}\frac{a_i \cap a_j}{a_i}, \tag{9}$$

a_i means the area of the ith box in the set of elements bounding boxes $\mathbf{e}$. In this work, $\mathbf{e}$ refers to the set of elements bounding boxes of the $\{logo, text\}$ or $\{embellishment\}$.

$$R_{ove} = R^{\{logo,text\}}_{ove} + R^{\{embellishment\}}_{ove} \tag{10}$$

The underlay, as a background element, is primarily used to emphasize or highlight another visual element. Thus, it is expected that an underlay should not appear alone but be overlaid by at least one other type of element (e.g., text or logo). To evaluate the degree to which underlay elements are functionally utilized, we define the underlay overlap degree metric R_{und}. This metric quantifies the extent to which underlay regions are overlapped by other content elements. Specifically, for each underlay element, we compute its maximal overlap ratio with elements of other types and then sum the results over all underlays. A higher R_{und} value indicates that underlays are more effectively used to support visible content. Although this formulation might seem to favor cases where underlays are closely fitted to overlaid elements, its actual purpose is to assess whether underlays are being effectively used to support other visual content. If an underlay is largely covered by meaningful elements such as text or logos, it indicates that the layout is making effective use of the underlay to enhance visual emphasis, which leads to a higher score. Conversely, underlays that remain mostly uncovered are considered less functional and result in lower scores. We adapt Eq. 9 to calculate R_{und} as follows:

$$R_{und} = \sum_{i \in \mathbf{e}_1} \max_{j \in \mathbf{e}_2} \frac{a_i \cap a_j}{a_i} \tag{11}$$

$$\mathbf{e}_1 = \{underlay\}$$

$$\mathbf{e}_2 = \{logo, text, embellishment\},$$

where a_i means the area of the ith box in $\mathbf{e}_1$, and $\mathbf{e}_1$ and $\mathbf{e}_2$ represent the set of elements.

The metric R_{ali} is used to demonstrate that elements in an aesthetic graphic layout tend to align in one dimension. We follow the equation in [3] to calculate the alignment distance of elements:

$$R_{ali} = \sum_{i=1}^{N} \min\left(\mathcal{D}x_i^l, \mathcal{D}x_i^c, \mathcal{D}x_i^r, \mathcal{D}y_i^t, \mathcal{D}y_i^c, \mathcal{D}y_i^b\right) \tag{12}$$

N represents the total number of predicted elements. $\mathcal{D}x_i^l, \mathcal{D}x_i^c, \mathcal{D}x_i^r, \mathcal{D}y_i^t, \mathcal{D}y_i^c, \mathcal{D}y_i^b$ represent the minimum distance between the ith bounding box and other bounding boxes in the dimensions of the left, horizontal midpoint, right, top, vertical midpoint, and bottom, respectively. $\mathcal{D}x_i^* \ (* = l, c, r)$ refers to [3] as:

$$\mathcal{D}x_i^* = \min_{j \neq i} \left|x_i^* - x_j^*\right| \tag{13}$$

$\mathcal{D}y_i^* \ (* = t, c, b)$ can be calculated similarly.

To enable computation of the conventional FID [16], which requires ground-truth layout annotations, the test set of 6,002 annotated layout-image pairs from the CGL-Dataset is utilized. For each image, the model generates a corresponding layout. FID is then computed between the distributions of real and generated layout features. Similar in form to Eq. 8, FID is calculated as follows:

$$\begin{aligned} \mathrm{FID} = & \left\|\mu_{\boldsymbol{l}_{GT}} - \mu_{\boldsymbol{l}_{pre}}\right\|_2^2 \\ & + \mathrm{Tr}\left(\Sigma_{\boldsymbol{l}_{GT}} + \Sigma_{\boldsymbol{l}_{pre}} - 2 \cdot \left(\Sigma_{\boldsymbol{l}_{GT}} \Sigma_{\boldsymbol{l}_{pre}}\right)^{\frac{1}{2}}\right) \end{aligned} \tag{14}$$

where $\mu_{\boldsymbol{l}_{GT}}$, $\Sigma_{\boldsymbol{l}_{GT}}$ and $\mu_{\boldsymbol{l}_{pre}}$, $\Sigma_{\boldsymbol{l}_{GT}}$ denote the mean and covariance of layout features extracted from the real and generated layouts, respectively.

VI. Experiments

In this section, we first introduce the implementation details of our experimental setup. Next, we present both quantitative and qualitative comparisons to demonstrate that our model achieves SOTA performance. We then provide visual examples to show how PDA-GAN effectively bridges the domain gap. Finally, we conduct ablation studies to evaluate the contribution of each component to the quality of the generated layouts, and extend our experiments to explore the effect of the training dataset, natural language-guided layout generation, and failure cases from alternative discriminator designs.

A. Implementation Details

We implement our model using PyTorch 1.7.1 and train it with the Adam optimizer [60]. The initial learning rate is set to 10^{-5} for the generator backbone and 10^{-4} for the remaining parts of the model. Training is conducted for 300 epochs with a batch size of 128, and all learning rates are reduced by a factor of 10 after 200 epochs. For fair comparison in the ablation studies, inpainted posters and product images are resized to 240×350 before being used as input.

We observe that, during training, the network is prone to bias towards source domain data. It might be due to the additional reconstruction loss for the source domain to supervise the generator of the model. Therefore, to balance the influence of the two domains, 8000 samples are randomly selected from the CGL-Dataset as the source domain data. In each epoch, the 8000 source domain samples are processed, and another 8000 samples of the target domain images are randomly selected. We refer to this choice of training data as Data I. If all the CGL-Dataset training images are used for comparison, we refer to it as Data II. In the following, if not clearly mentioned, PDA-GAN (CGL-GAN) is trained with Data I (Data II). The total training time for PDA-GAN on Data II is about 8 hours, while CGL-GAN on Data I takes approximately 43.5 hours, both utilizing 16 NVIDIA V100 GPUs.

B. Comparison with State-of-the-art Methods

Layout generation with image contents. We begin by conducting experiments to compare PDA-GAN with ContentGAN [4], Layoutprompter [32], and CGL-GAN, which are capable of generating image-aware layouts. Quantitative results can be seen from Tab. II. Our models, PDA-GAN and CGL-GAN, have demonstrated strong performance across the majority of metrics. Notably, PDA-GAN achieves the best results in most metrics, especially in the content-aware metrics, since PDA-GAN preserves the image color and texture details. For example, PDA-GAN outperforms ContentGAN, Layoutprompter, and CGL-GAN by 26.4%, 15.6%, and 6.21%, respectively, with respect to background complexity R_{com}. As shown in the first column in Fig. 5, compared with those by

TABLE II: **Comparison with image-aware methods.** Bold and underlined numbers denote the best and second best respectively. ↓ (or ↑) means the smaller (or bigger) value, the better.

Model	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$cFID\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$	$FID\downarrow$	$R_{occ}\uparrow$
ContentGAN [4]	45.59	17.08	1.143	26.30	0.0397	0.8626	<u>0.0071</u>	9.62	93.4
Layoutprompter [32]	39.77	<u>14.66</u>	0.840	25.55	0.2251	0.6786	**0.0068**	8.67	99.5
CGL-GAN(Ours)	<u>35.77</u>	15.47	<u>0.805</u>	<u>15.31</u>	**0.0233**	<u>0.9359</u>	0.0098	<u>5.10</u>	<u>99.6</u>
PDA-GAN(Ours)	**33.55**	**12.77**	**0.688**	**12.39**	<u>0.0290</u>	**0.9481**	0.0105	**4.98**	**99.7**

TABLE III: **User study.** P_e (P_b) represents the percentage of eligible-selected (best-selected) layouts. The symbol * denotes the professional group.

Model	$P_e\uparrow$	$P_b\uparrow$	$P_e^*\uparrow$	$P_b^*\uparrow$
ContentGAN [4]	18.89	19.36	18.25	16.58
Layoutprompter [32]	26.25	20.12	24.58	18.24
CGL-GAN	26.12	28.36	26.45	26.46
PDA-GAN	**28.74**	**32.16**	**30.72**	**38.72**

TABLE IV: **Comparison with image-agnostic methods.** LT and VTN represent LayoutTransformer and LayoutVTN, respectively.

Model	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$cFID\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$	$FID\downarrow$
LT [31]	40.92	21.08	1.310	27.24	0.0156	0.9516	0.0049	6.25
VTN [15]	41.77	22.21	1.323	30.14	0.0130	**0.9698**	0.0047	7.13
LDM [26]	41.20	27.91	1.792	32.36	0.0146	0.9532	0.0032	5.02
LD [27]	40.56	27.36	1.772	28.76	**0.0116**	0.9624	**0.0028**	**4.28**
CGL-GAN	35.77	15.47	0.805	15.31	0.0233	0.9359	0.0098	5.10
PDA-GAN	**33.55**	**12.77**	**0.688**	**12.39**	0.0290	0.9481	0.0105	4.98

ContentGAN and CGL-GAN, bounding boxes of text elements generated by PDA-GAN are more likely to appear in simple background areas, which improves the readability of the text information. As shown in the second and third columns, when the background of the text element is complex, PDA-GAN will generate an underlay bounding box to replace the complex background to enhance the readability of text information.

Compared to ContentGAN, Layoutprompter, and CGL-GAN, PDA-GAN reduces the subject occlusion degree R_{shm} by 25.2%, 12.9%, and 17.5%, respectively. From the three middle columns of Fig. 5, for ContentGAN or CGL-GAN, the presentation of the subject content information is largely affected since the generated layout bounding boxes would inevitably occlude subjects. In particular, it should be noted that when the layout bounding box occludes the critical regions of the subject, such as the human head or face, the visual effect of the poster will be unpleasant, taking the image in row-3-column-6 as an example. In contrast, layout bounding boxes generated by PDA-GAN avoid subject regions nicely, thus the generated posters better express the information of subjects and layout elements.

Meanwhile, the product occlusion degree R_{sub} of PDA-GAN performance surpasses ContentGAN, Layoutprompter, and CGL-GAN by 39.8%, 18.1%, and 14.5%, respectively. The three rightmost columns in Fig. 5 are the heat maps of the attention of each pixel to the product in the image. We get attention maps of product images (queried by their category tags extracted on product pages) by CLIP [57], [58]. Compared with ContentGAN and CGL-GAN, PDA-GAN generates layout bounding boxes on the region with lower thermal values to avoid occluding products. For example, in the seventh column, the layout bounding box generated by PDA-GAN effectively avoids the region with high thermal values of the product, which enables the hoodie information of the product to be fully displayed.

Compared to ContentGAN, LayoutPrompter, and CGL-GAN, PDA-GAN reduces the cFID score by 52.9%, 51.5%, and 19.1%, respectively, indicating better semantic harmony between layout and image content. Even in terms of the conventional FID, PDA-GAN achieves lower scores than the above baselines, demonstrating that it better captures the distribution of real poster layouts. The above quantitative and qualitative comparisons of models demonstrate that PDA-GAN improves the relationship modeling between graphic layouts and image contents, including color and texture details.

To provide a more comprehensive evaluation beyond standard quantitative metrics, we conduct a user study, as summarized in Tab. III. A total of 120 test samples are randomly selected, each consisting of a product image and four corresponding layouts generated by ContentGAN, LayoutPrompter, CGL-GAN, and PDA-GAN. The participants include two groups: 10 professional designers and 20 novice designers. Each participant is asked to identify both the eligible and the best layout among the four candidates for each sample. We report the eligible selection rate (P_e) and best selection rate (P_b), defined as the proportion of votes received by each model relative to the total votes. Results show that PDA-GAN consistently outperforms other methods, particularly with a significantly higher proportion of best-selected (P_b) layouts.

Layout generation without image contents. We also compare our method with image-agnostic approaches, including LayoutTransformer [31], LayoutVTN [15], LayoutDM [26], and LayoutDiffusion [27]. As shown in Tab. IV, these image-agnostic methods perform well on graphic metrics. However, our models significantly outperform them on content-aware metrics. Specifically, PDA-GAN surpasses LayoutTransformer, LayoutVTN, LayoutDM, and LayoutDiffusion by 18.0%, 19.7%, 18.6%, and 17.3%, respectively, in terms of background complexity R_{com}. This is because the image-agnostic methods only model the relationships between layout elements, without considering the image content. These image-agnostic methods tend to generate bounding boxes for text elements in areas with complex backgrounds (as shown in the first two rows and the leftmost three columns of Fig. 5), which reduces the readability of the text information. Furthermore, compared with LayoutTransformer, LayoutVTN, LayoutDM, and LayoutDiffusion, PDA-GAN reduces R_{shm} by 39.4%, 42.5%, 54.2%, and 53.3%, and reduces R_{sub} by 47.5%, 48.0%, 61.6%, and 61.2%, respectively. The rightmost six

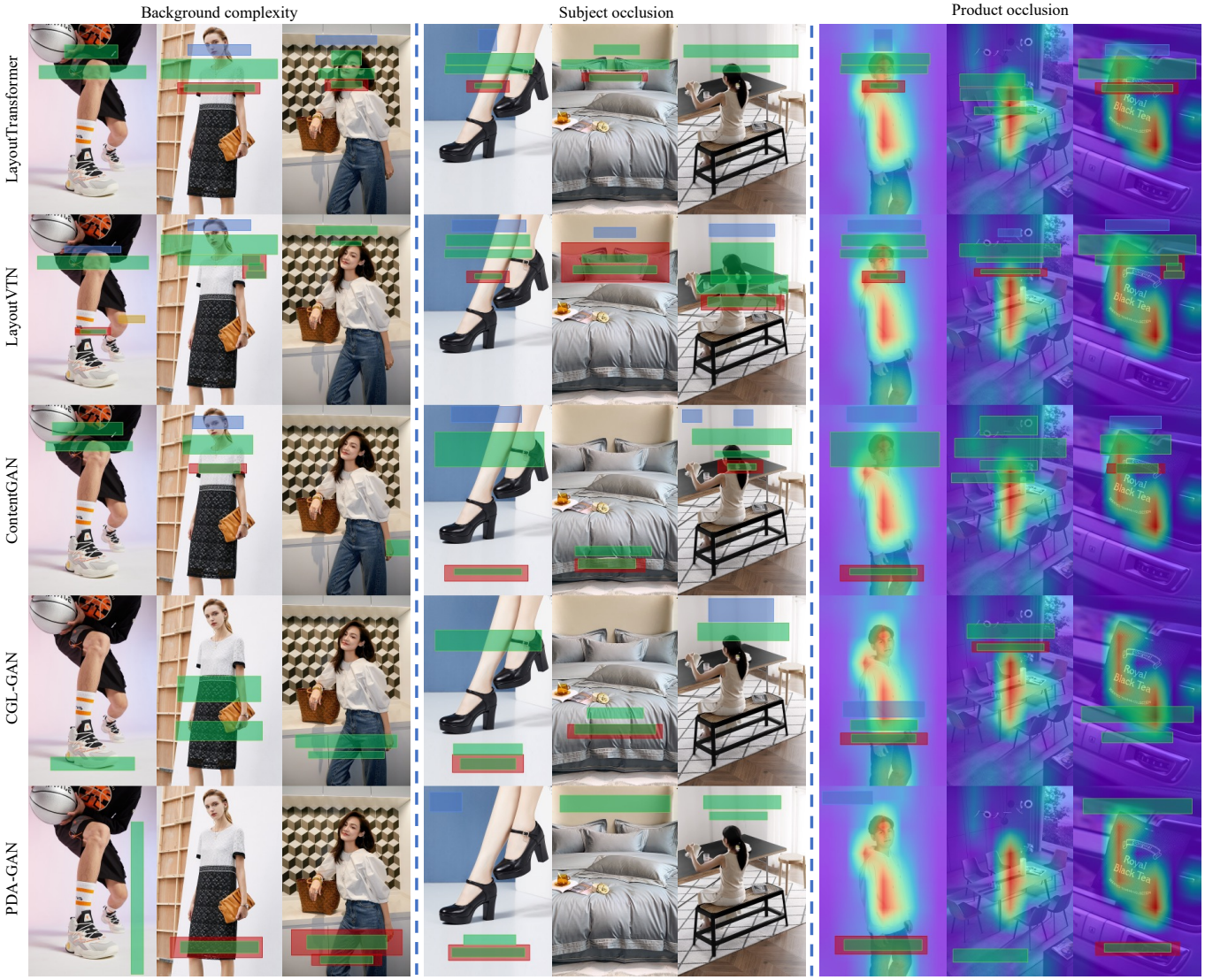


Fig. 5: **Qualitative evaluation for different models.** Layouts in each column are conditioned on the same image, while those in each row are generated by the same model. This figure provides a qualitative comparison and analysis of different models from three perspectives: background complexity of text elements, subject overlap, and product overlap attention maps.

TABLE V: **Comprehensive comparison between CGL-GAN and PDA-GAN.** Data I and Data II contain 8,000 and 54,546 source domain samples, respectively. ✓indicates the experiment configuration. The symbol "-" indicates that the model cannot complete the layout generation task since the generated element bounding boxes overlap with each other severely.

Model	Data I	Data II	Gaussian Blur	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$	$R_{occ}\uparrow$
CGL-GAN	✓			33.85	13.88	0.766	0.0299	0.9351	0.0139	**99.7**
CGL-GAN	✓		✓	-	-	-	2.5826	-	-	-
CGL-GAN		✓	✓	35.77	15.47	0.805	**0.0233**	0.9359	**0.0098**	99.6
PDA-GAN	✓		✓	36.41	19.48	1.044	0.0244	0.9384	0.0091	99.7
PDA-GAN		✓	✓	**30.63**	18.14	0.833	0.0589	0.9302	0.0105	99.6
PDA-GAN	✓			33.55	**12.77**	**0.688**	0.0290	**0.9481**	0.0105	**99.7**

columns in Fig. 5 show that image-agnostic methods generate layout bounding boxes that randomly occlude subject and product areas. Such occlusions degrade the visibility and presentation of both subject and product content.

More comparisons with CGL-GAN. As shown in the first and last rows of Tab. V, PDA-GAN consistently outperforms CGL-GAN across all metrics under identical configurations, validating the effectiveness of the proposed PD. Unlike CGL-GAN, PDA-GAN replaces both the Gaussian blur pre-processing and the heavy discriminator network with a lightweight PD module (only 332,545 parameters, less than 2% of the 22,575,841 parameters in CGL-GAN), significantly reducing memory and computation cost. To enable a fair comparison, PDA-GAN is also trained under the same con-

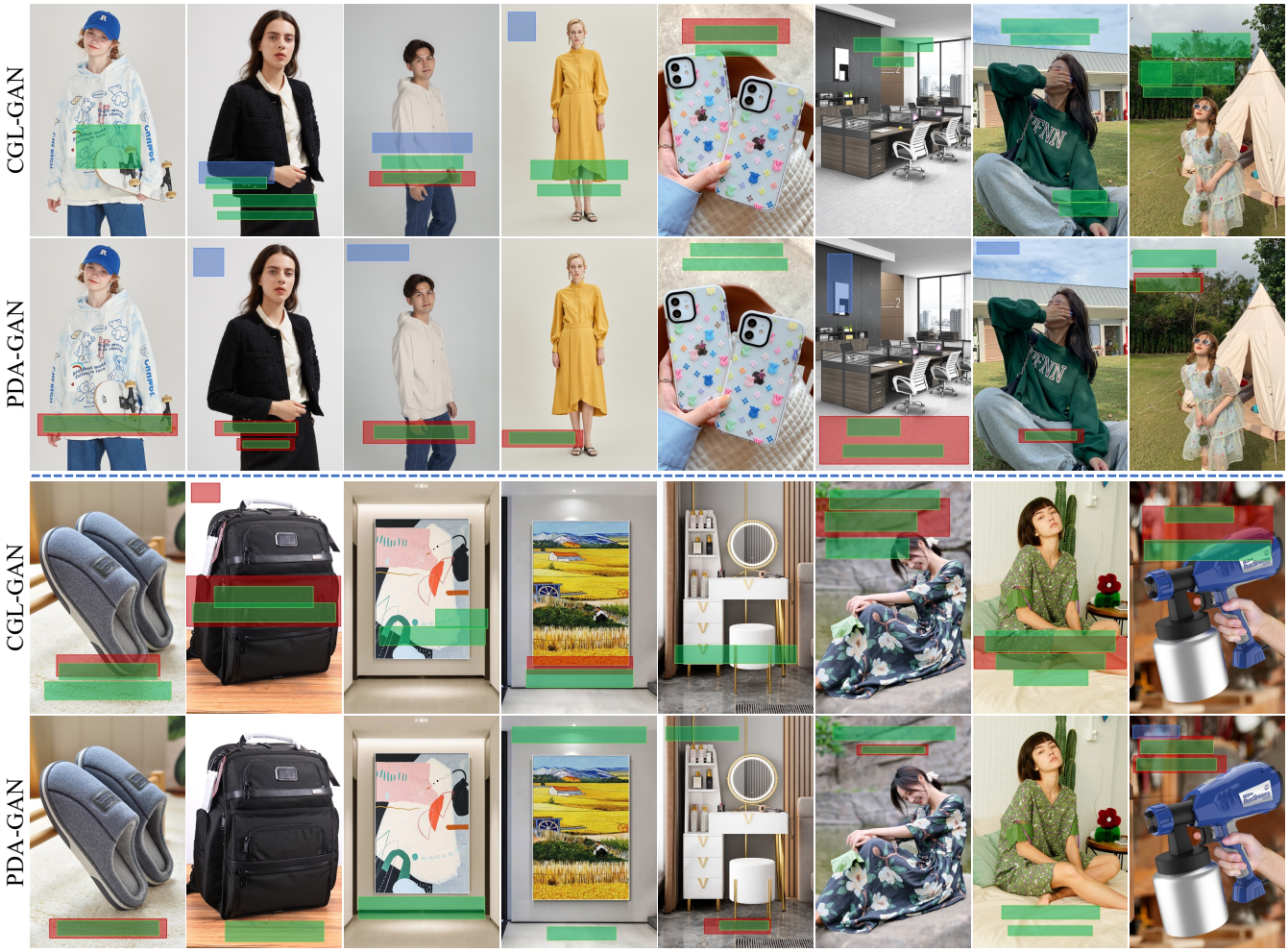


Fig. 6: More qualitative comparisons between CGL-GAN and PDA-GAN.

figuration as CGL-GAN. As shown in the middle rows of Tab. V, CGL-GAN trained on Data I with Gaussian blur fails to produce usable layouts, resulting in extremely high R_{ove} due to severely overlapping layout elements. In contrast, PDA-GAN under the same setting generates reasonable and well-structured layouts. These results highlight the stronger generalization ability of PDA-GAN across different domain conditions. Intuitively, Gaussian blur can help narrow the domain gap, but it also causes loss of image color and texture details. As shown in the fourth and sixth rows of Tab. V, PDA-GAN without Gaussian blur achieves the best performance across all content-aware metrics. This result shows that PDA-GAN can eliminate the domain gap without relying on Gaussian blur, while preserving image details for generating high-quality layouts.

As illustrated in the sixth and eighth columns of the first two rows in Fig. 6, compared with CGL-GAN, PDA-GAN generates text bounding boxes with a simpler background. It is interesting to observe from the first two rows of Fig. 6 that when PDA-GAN generates boxes among complex backgrounds, it tends to additionally generate an underlay bounding box which covers the complex background to ensure readability of the text information. The last two rows show that layouts generated by PDA-GAN can effectively avoid the subject area, and then can generate posters that better express the information of subjects and layout elements. Both the above quantitative and qualitative evaluations demonstrate that PDA-GAN can capture the subtle interaction between image contents and graphic layouts and achieve the SOTA performance.

C. Eliminating Domain Gap

As shown in Fig. 7, we randomly select four clean product images $\boldsymbol{x}_{img}$ from the target domain data and add graphic layout elements to these images to create advertising posters $\boldsymbol{x}_{pst}$. Inpainting the regions of elements in posters to obtain inpainted images $\boldsymbol{x}_{pst}^{inp}$. Due to inpainted areas, there is a domain gap between $\boldsymbol{x}_{img}$ (target domain) and $\boldsymbol{x}_{pst}^{inp}$ (source domain).

To demonstrate that PDA-GAN can effectively bridge the domain gap, we input $\boldsymbol{x}_{pst}^{inp}$ and $\boldsymbol{x}_{img}$ to CGL-GAN and PDA-GAN to generate layouts. The mean difference values of the shallow-level feature maps, fusion feature maps, and deep-level feature maps generated by CGL-GAN between $\boldsymbol{x}_{pst}^{inp}$ and $\boldsymbol{x}_{img}$ of the above four samples as input are 0.0610,

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2025.3602846

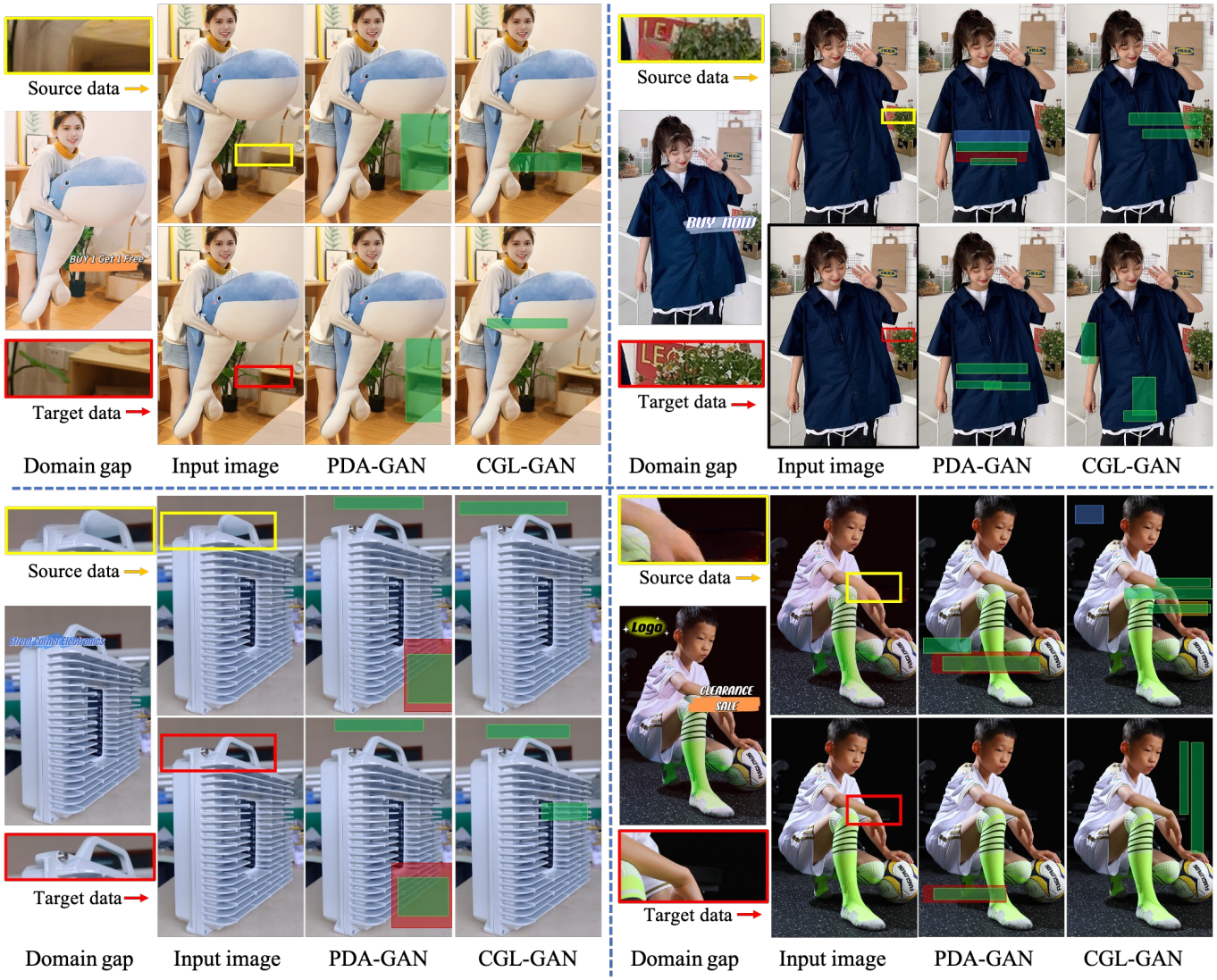


Fig. 7: **Layouts generated by different models using source and target domain data.** Inpainted images (source domain) and clean images (target domain) are both fed into PDA-GAN and CGL-GAN. The results produced by PDA-GAN are relatively consistent, indicating that our method achieves better feature alignment between the two domains.

0.1289, and 0.1263. The corresponding mean values calculated by PDA-GAN are 0.0293, 0.0726, and 0.1160, respectively. Compared with CGL-GAN, PDA-GAN has less difference in the generated feature maps between the source and target domain data.

From the perspective of generated results, layouts generated by CGL-GAN under different input domains (i.e., clean images $\boldsymbol{x}_{img}$ vs. inpainted images $\boldsymbol{x}_{pst}^{inp}$) exhibit noticeable inconsistencies. Specifically, CGL-GAN tends to place layout elements in distorted or blurred regions of the inpainted images, likely because the annotated bounding boxes in the source domain have all been inpainted during training. In contrast, PDA-GAN produces layouts that are visually more consistent across these two inputs. To validate the effectiveness of the pixel-level discriminator in bridging the domain gap, we further compute FID between the layouts generated from $\boldsymbol{x}_{img}$ and $\boldsymbol{x}_{pst}^{inp}$. PDA-GAN achieves a significantly lower FID score of 12.67 compared to 15.34 for CGL-GAN, indicating enhanced robustness to domain shifts. This result, along with the feature-level comparison, demonstrates that PDA-GAN can effectively eliminate the domain gap caused by inpainting.

D. Ablations

Effects of the discriminator at the pixel level. We first compare our PD with a global discriminator (GD) that only predicts one real or fake probability as in classical GAN. The abbreviation GD in Tab. XII indicates the global discriminator strategy. When the weight of GD loss (γ in Eq. 4) is more than 0.01, the model cannot complete the layout generation task, indicated by the symbol $-$, since the R_{ove} value is too high. From the statistics in Tab. XII, our PD outperforms GD on all metrics.

Second, we compare PD with the PatchGAN strategy [12]. The patch size in the Tab. VII refers to the dimension of the output map, which is then compared with the correspondingly resized ground-truth white-patch map during training. We train

TABLE VI: **Ablation study with global discriminators.** W refers to the weight of GD (or PD) module loss in the training process.

Model-W	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
GD-6.0	-	-	-	9.0000	-	-
GD-1.0	-	-	-	8.9995	-	-
GD-0.01	-	-	-	4.7764	-	-
GD-0.001	34.41	13.78	0.749	0.0327	0.9299	0.0110
GD-0.0001	34.77	14.62	0.777	0.0345	0.9234	0.0122
GD-0.0	34.07	15.13	0.800	0.0350	0.9259	0.0108
PD-6.0	**33.55**	**12.77**	**0.688**	**0.0290**	**0.9481**	**0.0105**

TABLE VII: **Ablation study with PatchGAN-based methods.** The input image height and width are 320 and 240 respectively.

Patch size	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
12*8	-	-	-	0.9288	-	-
24*16	33.67	16.00	0.844	0.0438	0.9407	**0.0075**
44*30	34.03	13.02	0.752	**0.0284**	0.9377	0.0119
88*60	**32.65**	13.35	0.735	0.0325	0.9173	0.0094
350*240	33.55	**12.77**	**0.688**	0.0290	**0.9481**	0.0105

these models with γ in Eq. 4 set to 6. The values of quantitative metrics listed in Tab. VII also confirm the advantage of the pixel-level discriminator. These experiments demonstrate that, due to the discrepancy by inpainting at the pixel level, the model might need to eliminate the domain gap at the pixel level. Additionally, the pixel-level strategy in the last row of Tab. VII can be considered to be the most fine-grained approach at the patch level.

Effects of PD with different level feature maps. In our model, PD is connected to the shallow-level feature maps of the first residual block. We now investigate PD performance when utilizing the deep-level feature from the fourth residual block and the fused feature (fusion of feature maps from the first to fourth residual blocks) in multi-scale CNN. As shown in Tab. VIII, discriminating with shallow feature maps in PDA-GAN can achieve better results in both content-aware and graphic metrics on average. These results further validate the effectiveness of our PD design. Intuitively, bridging the domain gap at an early stage in the network may benefit subsequent processing within the model.

Effects of PD with different architectures. To validate the robustness and adaptability of the proposed PD, we conduct ablation studies focusing on two architectural aspects: kernel size and network depth. First, we compare PD modules using convolutional kernels of size 3×3, 5×5, 7×7, as well as a hybrid configuration that stacks layers with 3, 5, and 7-sized kernels. As shown in Tab. IX, the performance remains stable across different configurations, with the 3×3 kernel achieving slightly better results on most metrics. In addition, we investigate the impact of network depth by varying the number of transposed convolution layers in PD (2, 3, 6, and 9 layers). The results, summarized in Tab. X, indicate that performance is comparable across different depths. These results confirm that the proposed PD design is compact and effective, with a lightweight 3×3 kernel and a relatively shallow network being sufficient to handle the domain discrepancies introduced by inpainting.

TABLE VIII: **Quantitative ablation study on different level feature maps for the pixel-level discriminator.**

Feature map	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
deep level	34.22	13.97	0.770	0.0396	0.9366	0.0118
fusion	35.36	14.54	0.817	0.0310	**0.9513**	0.0117
shallow level	**33.55**	**12.77**	**0.688**	**0.0290**	0.9481	**0.0105**

TABLE IX: **Quantitative ablation study on PD with different kernel sizes.** 3, 5, and 7 denote convolutional kernel sizes of 3×3, 5×5, and 7×7, respectively, while 'fusion' indicates a hybrid design combining all three.

Size	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
5	33.09	13.66	0.720	0.0349	0.8891	0.0086
7	**31.69**	14.73	0.743	0.0431	0.9185	**0.0081**
fusion	32.30	13.25	0.747	0.0333	0.9316	0.0095
3	33.55	**12.77**	**0.688**	**0.0290**	**0.9481**	0.0105

TABLE X: **Quantitative ablation study on different level feature maps for the pixel-level discriminator.** 2, 3, 6, and 9 denote the number of transposed convolutional layers used in the PD module.

Layers	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
2	**32.37**	13.48	**0.687**	0.0360	0.9151	0.0110
6	32.97	15.25	0.792	0.0318	0.9040	0.0130
9	34.07	13.80	0.743	**0.0279**	0.9295	**0.0086**
3	33.55	**12.77**	0.688	0.0290	**0.9481**	0.0105

TABLE XI: **Ablation study with different label smoothing choices.** The first row is the model without label smoothing. Two-side: set 0 to 0.2 and 1 to 0.8; one-source: set 1 to 0.8; and 0ne-target: set 0 to 0.2.

smoothing	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
without	33.61	14.04	0.718	0.0346	0.9188	0.0106
two-side	33.66	14.67	0.794	0.0334	0.9297	0.0098
one-source	**32.20**	15.23	0.799	0.0431	0.9234	**0.0085**
one-target	33.55	**12.77**	**0.688**	**0.0290**	**0.9481**	0.0105

Effects of label smoothing. For the ground truth white-patch map input to the discriminator, the two-side label smoothing means we set 0 to 0.2 and 1 to 0.8, the one-source label smoothing means we only set 1 to 0.8, and the one-target label smoothing means we only set 0 to 0.2. The first row in Tab. XI means the model without label smoothing. Tab. XI shows that the model with one-target label smoothing performs better in all metrics than without label smoothing, demonstrating its effectiveness. In addition, the effects of two-side or one-source label smoothing are not as good as one-target label smoothing on average. In contrast, both two-side and one-source label smoothing perform worse than one-target label smoothing.

Effects of PD. Compared to the model without the PD module in the first row of Tab. XII, under the same configuration, the model with the PD module achieves better results in all metrics. Benefiting from the PD module, which effectively eliminates the domain gap, as demonstrated in Sec. VI-C, the model with the PD module can generate high-quality, image-aware graphic layouts for advertising posters.

Replacing DETR with deformable DETR. We further evalu-

Fig. 8: **Impact of Gaussian blur.** Layouts in each row are generated using the same input image, while those in each column correspond to different input samples. "PDA-GAN with Gaussian blur" indicates that the input data is preprocessed with Gaussian blurring. The blue-numbered boxes in the middle show enlarged views of the regions marked with yellow numbers. The left part of the vertical dotted line displays the input images, while the right part displays the corresponding product attention heatmaps.

TABLE XII: **Quantitative ablation study on PD.** Ours' refers to our model of PDA-GAN without the PD module.

Model	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
Ours'	34.07	15.13	0.800	0.0350	0.9259	0.0108
Ours	**33.55**	**12.77**	**0.688**	**0.0290**	**0.9481**	**0.0105**

TABLE XIII: **Quantitative ablation study on different backbones.** D-DETR denotes deformable DETR.

Backbone	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
D-DETR	**32.48**	13.03	0.698	0.1085	0.9423	**0.0078**
DETR	33.55	**12.77**	**0.688**	**0.0290**	**0.9481**	0.0105

TABLE XIV: **Quantitative ablation study on Gaussian blur.** Ours* refers to the PDA-GAN model with Gaussian blur applied to the input image.

Model	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
Ours*	36.71	20.14	1.036	0.0475	0.9376	**0.0068**
Ours	**33.55**	**12.77**	**0.688**	**0.0290**	**0.9481**	0.0105

ate the generalization ability of our approach by replacing the DETR-based layout generator with deformable DETR, a multi-scale transformer model known for its better convergence and object localization. As shown in Tab. XIII, our method maintains strong performance, with DETR yielding slightly better results on several layout accuracy metrics. Since the PD module performs domain adaptation on shallow visual features, feeding only high-level features into the transformer can be more effective, as the domain gap has already been eliminated at earlier stages. These results validate that our method generalizes well across different detection backbones and that the PD module remains effective with both DETR and deformable DETR.

Effect of Gaussian blur. Based on the PDA-GAN model, we quantitatively and qualitatively analyze the impact of Gaussian blur on layout generation, as shown in Tab. XIV and Fig. 8. Applying Gaussian blur leads to an increase in R_{com} from 33.55 to 36.71. As shown in boxes 2, 6, and 10 of Fig. 8, the model with Gaussian blur tends to generate text bounding boxes with more complex backgrounds, which reduces text readability. In contrast, boxes 7 and 11 illustrate that the model without Gaussian blur generates text bounding boxes with simpler backgrounds or introduces underlay bounding boxes to replace complex backgrounds.

Tab. XIV shows that R_{shm} and R_{sub} of the model with Gaussian blur increase from 12.77 to 20.14 and from 0.688 to 1.036, respectively. As shown in Fig. 8, the model with

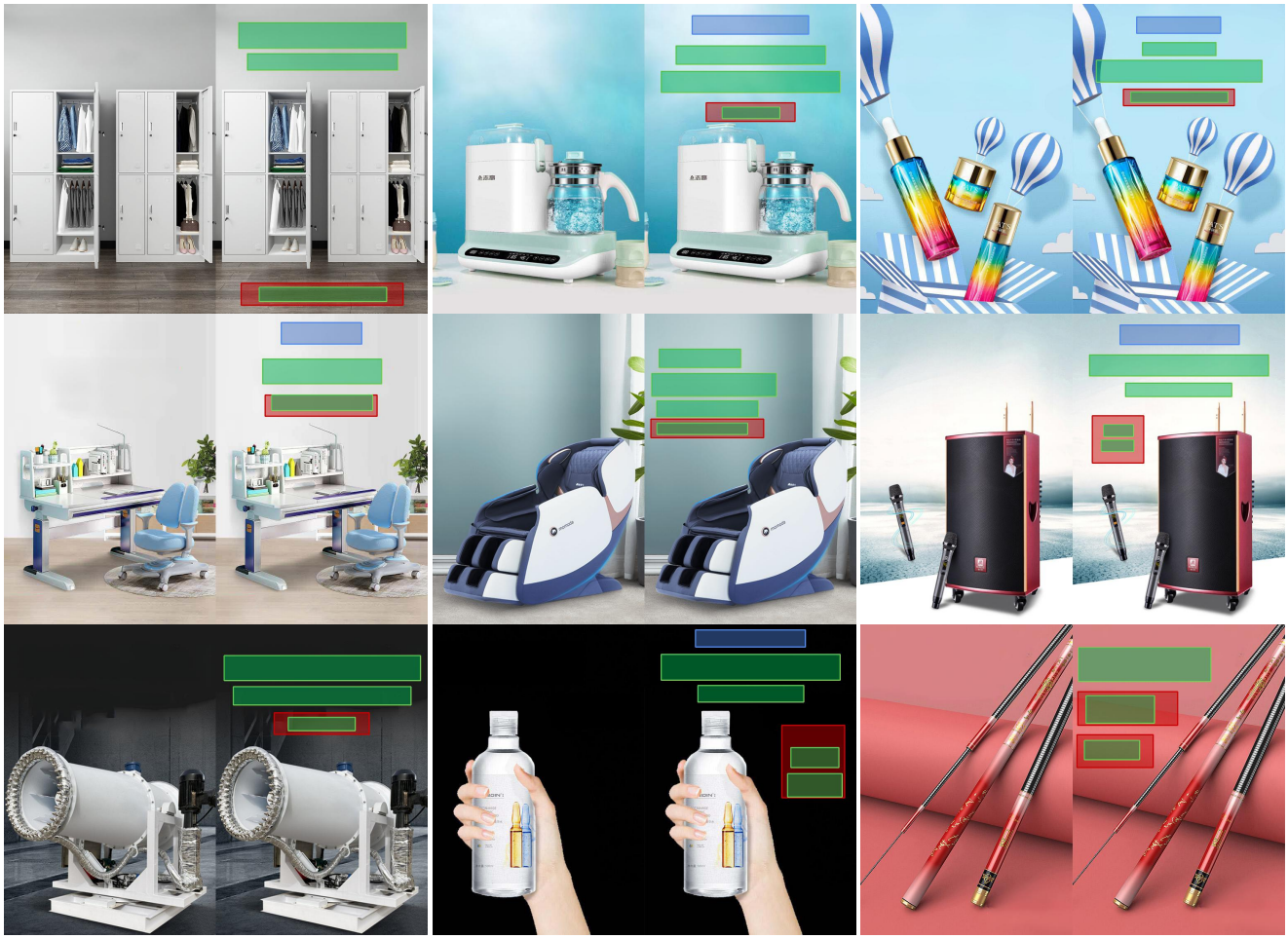


Fig. 9: **Generalization ability of PDA-GAN on PKU-Dataset [33].**

Gaussian blur tends to generate layout bounding boxes that occlude subject or product regions. Such layouts diminish the visibility of subjects and the clarity of layout elements in advertising posters. These quantitative and qualitative results demonstrate that the loss of image details caused by Gaussian blur degrades the quality of the generated image-aware graphic layouts.

E. Extension Experiments

Effect of training dataset. To further validate the advantages of the CGL-Dataset for image-aware layout generation, we conduct additional experiments using the publicly available PKU-Dataset [33], which, as introduced in Sec. III, includes a broader variety of poster types and layout styles. Specifically, we train the PDA-GAN model separately on the CGL-Dataset and the PKU-Dataset, and evaluate both models on 1,000 clean product images. As shown in Tab. XV, the PDA-GAN trained on the CGL-Dataset consistently outperforms the model trained on the PKU-Dataset across all evaluation metrics. This result highlights the benefits of the CGL-Dataset, which, compared to the PKU-Dataset, offers a larger scale, more diverse element categories, and better alignment with the needs of image-aware layout generation.

TABLE XV: **Comparison of PDA-GAN trained on different datasets.**

Dataset	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$
PKU [33]	33.88	18.03	0.881	0.0445	0.9411	0.0127
CGL(ours)	**33.55**	**12.77**	**0.688**	**0.0290**	**0.9481**	0.0105

Generality to other poster types. To further demonstrate the generalizability of the PDA-GAN, we evaluate the model's performance on the PKU-Dataset test set [33], as shown in Fig. 9. This dataset contains a diverse range of poster types with varying design styles and layout structures. The generated layouts effectively avoid interference with the main subject areas of the posters, ensuring that both the background image and layout elements are displayed harmoniously. This demonstrates that PDA-GAN can successfully generalize to new poster types and adapt to different layout styles, further validating its robustness and flexibility.

Extension to natural language-guided layout generation with LLMs. Although originally designed as a discriminator in a GAN-based pipeline, the PD module can be flexibly integrated with other types of generation backbones, including LLM-based and diffusion-based models. To explore this adaptability, we extend our framework to a language-guided layout

generation setting, where the PD module is integrated with LLM-based components. In this extension, CLIP [57] is used to extract textual features from user-provided natural language prompts, which are then fused with multi-scale image features. These combined features are passed through a transformer encoder, followed by fully connected layers, to generate layout predictions. The input language prompts are structured as simple declarative sentences, such as "generate three elements" or "generate text and underlay". To enhance generalization, we use GPT [61] to augment the input prompts, generating multiple semantically similar variants during training. This allows the model to better capture the diversity of natural language expressions. As illustrated in Fig. 10, the model is capable of generating image-aware layouts that are consistent with both the semantic content of the image and a broad range of natural language instructions, including those that specify the number and categories of layout elements. These results demonstrate that the proposed PD module is a flexible component that can be adapted to a variety of generation frameworks, including those guided by natural language.

Analysis of failure cases from alternative discriminator designs. To complement our evaluation, we analyze the failure cases of alternative discriminator designs, specifically those using either global or patch-level supervision, trained with identical configurations as our full model. Quantitative results are presented in the first rows of Tab. XII and Tab. VII, where both alternatives yield significantly higher layout overlap ratios. As shown in the first and second rows of Fig. 11, the generated layouts exhibit severe element overlap and visual clutter, making them unsuitable for practical poster design applications. These examples demonstrate that directly applying global or patch-level discriminators fails to effectively address the domain gap, and thus cannot produce usable image-aware layouts. In contrast, our pixel-level discriminator enables fine-grained domain alignment at the early visual stage, resulting in high-quality layouts that are better aligned with image content.

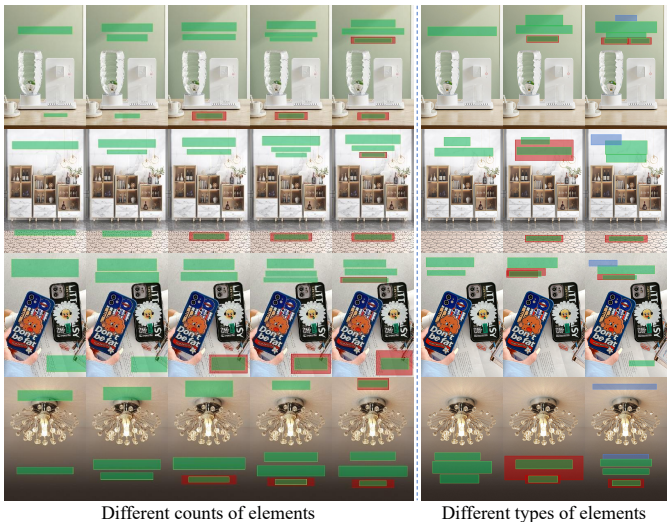


Fig. 10: **Language-guided layout generation with varying element counts and types.** Each row corresponds to the same background image. The left part shows layouts generated from prompts specifying the number of elements (e.g., "generate three elements", 2 to 6 from left to right), while the right part shows layouts guided by prompts specifying element types (e.g., "generate text and underlay").

Fig. 11: **Failure cases.** The first to third rows show layouts generated by models with the global discriminator (GD), patch discriminator (PatchD), and our PD module, respectively.

VII. Conclusion

In this paper, we focus on generating image-aware graphic layouts for advertising posters. We introduce a novel generative framework, PDA-GAN, designed to first bridge the domain gap and then model the relationship between image content and layouts. For the image-aware layout generation task, we contribute a large advertising graphic layout dataset and propose three novel content-aware metrics. Both quantitative and qualitative evaluations demonstrate that our method achieves state-of-the-art performance and generates high-quality image-aware graphic layouts for posters. To support further research in graphic layout generation, we will release our model code and dataset to the community. In the future, we plan to explore how to better integrate user constraints, such as element categories and coordinates, and enhance layout generation diversity. Additionally, we aim to develop an automated system for end-to-end generation of high-quality posters directly from product images.

Acknowledgments

We sincerely thank the anonymous reviewers for their professional and constructive comments, which have helped us improve the quality and clarity of this paper. Weiwei Xu is partially supported by "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2023C01181). This work is supported by Alibaba Group through the Alibaba Innovation Research Program, the State Key Lab of CAD&CG, and the Information Technology Center, Zhejiang University.

REFERENCES

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Annual Conference on Neural Information Processing Systems*, 2014, pp. 2672–2680.

[2] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu, "Layoutgan: Generating graphic layouts with wireframe discriminators." ICLR, 2019.

[3] J. Li, J. Yang, J. Zhang, C. Liu, C. Wang, and T. Xu, "Attribute-conditioned layout GAN for automatic graphic design," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 10, pp. 4039–4048, 2021.

[4] X. Zheng, X. Qiao, Y. Cao, and R. W. H. Lau, "Content-aware generative modeling of graphic design layouts," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 133:1–133:15, 2019.

[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV (1)*, ser. Lecture Notes in Computer Science, vol. 12346. Springer, 2020, pp. 213–229.

[6] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," in *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022*. IEEE, 2022, pp. 3172–3182.

[7] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *ICCV*. IEEE Computer Society, 2015, pp. 2452–2460.

[8] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *ICCV*. IEEE Computer Society, 2015, pp. 4068–4076.

[9] H. Zhao, S. Zhang, G. Wu, J. M. F. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," in *NeurIPS*, 2018, pp. 8568–8579.

[10] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," *CoRR*, vol. abs/2010.03978, 2020.

[11] M. Jaritz, T. Vu, R. de Charette, É. Wirbel, and P. Pérez, "Cross-modal learning for domain adaptation in 3d semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1533–1544, 2023.

[12] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*. IEEE Computer Society, 2017, pp. 5967–5976.

[13] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu, "Layoutgan: Generating graphic layouts with wireframe discriminators," *CoRR*, vol. abs/1901.06767, 2019.

[14] A. A. Jyothi, T. Durand, J. He, L. Sigal, and G. Mori, "Layoutvae: Stochastic scene layout generation from a label set." ICCV, 2019, pp. 9894–9903.

[15] D. M. Arroyo, J. Postels, and F. Tombari, "Variational transformer networks for layout generation," in *CVPR*, 2021, pp. 13 642–13 652.

[16] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[17] D. Horita, N. Inoue, K. Kikuchi, K. Yamaguchi, and K. Aizawa, "Retrieval-augmented layout transformer for content-aware layout generation," in *CVPR*, 2024, pp. 67–76.

[18] M. Zhou, C. Xu, Y. Ma, T. Ge, Y. Jiang, and W. Xu, "Composition-aware graphic layout GAN for visual-textual presentation designs," in *IJCAI*. ijcai.org, 2022, pp. 4995–5001.

[19] C. Xu, M. Zhou, T. Ge, Y. Jiang, and W. Xu, "Unsupervised domain adaption with pixel-level discriminator for image-aware layout generation," in *CVPR*. IEEE, 2023, pp. 10 114–10 123.

[20] C. E. Jacobs, W. Li, E. Schrier, D. Bargeron, and D. Salesin, "Adaptive grid-based document layout," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 838–847, 2003.

[21] T. Kanungo and S. Mao, "Stochastic language models for style-directed layout analysis of document images," *IEEE Trans. Image Process.*, vol. 12, no. 5, pp. 583–596, 2003.

[22] R. Kumar, J. O. Talton, S. Ahmad, and S. R. Klemmer, "Bricolage: example-based retargeting for web design," in *Proceedings of the International Conference on Human Factors in Computing Systems*, 2011, pp. 2197–2206.

[23] Y. Cao, A. B. Chan, and R. W. H. Lau, "Automatic stylistic manga layout," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 141:1–141:10, 2012.

[24] P. O'Donovan, A. Agarwala, and A. Hertzmann, "Learning layouts for single-pagegraphic designs," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 8, pp. 1200–1213, 2014.

[25] R. Hedjam, H. Z. Nafchi, M. Kalacska, and M. Cheriet, "Influence of color-to-gray conversion on the performance of document image binarization: Toward a novel optimization problem," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3637–3651, 2015.

[26] N. Inoue, K. Kikuchi, E. Simo-Serra, M. Otani, and K. Yamaguchi, "Layoutdm: Discrete diffusion model for controllable layout generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 167–10 176.

[27] J. Zhang, J. Guo, S. Sun, J.-G. Lou, and D. Zhang, "Layoutdiffusion: Improving graphic layout generation by discrete diffusion probabilistic models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7226–7236.

[28] H. Lee, L. Jiang, I. Essa, P. B. Le, H. Gong, M. Yang, and W. Yang, "Neural design network: Graphic layout generation with constraints." ECCV, 2020, pp. 491–506.

[29] C. Yang, W. Fan, F. Yang, and Y. F. Wang, "Layouttransformer: Scene layout generation with conceptual and spatial diversity." CVPR, 2021, pp. 3732–3741.

[30] K. Kikuchi, E. Simo-Serra, M. Otani, and K. Yamaguchi, "Constrained graphic layout generation via latent optimization." ACM Multimedia Conference, 2021, pp. 88–96.

[31] K. Gupta, J. Lazarow, A. Achille, L. Davis, V. Mahadevan, and A. Shrivastava, "Layouttransformer: Layout generation and completion with self-attention," in *ICCV*. IEEE, 2021, pp. 984–994.

[32] J. Lin, J. Guo, S. Sun, Z. Yang, J.-G. Lou, and D. Zhang, "Layoutprompter: awaken the design ability of large language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 43 852–43 879, 2023.

[33] H. Hsu, X. He, Y. Peng, H. Kong, and Q. Zhang, "Posterlayout: A new benchmark and approach for content-aware visual-textual presentation layout," in *CVPR*. IEEE, 2023, pp. 6018–6026.

[34] D. Majumdar and V. P. Namboodiri, "Unsupervised domain adaptation of deep object detectors," in *ESANN*, 2018.

[35] S. Nagesh, S. Rajesh, A. Baig, and S. Srinivasan, "Domain adaptation for object detection using SE adaptors and center loss," *CoRR*, vol. abs/2205.12923, 2022.

[36] Y. Zhang and B. D. Davison, "Domain adaptation for object recognition using subspace sampling demons," *Multim. Tools Appl.*, vol. 80, no. 15, pp. 23 255–23 274, 2021.

[37] J. Zhang, J. Huang, Z. Tian, and S. Lu, "Spectral unsupervised domain adaptation for visual recognition," in *CVPR*. IEEE, 2022, pp. 9819–9830.

[38] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *CVPR*. IEEE Computer Society, 2017, pp. 95–104.

[39] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *AAAI*. AAAI Press, 2018, pp. 3934–3941.

[40] J. Gao, T. Zhang, and C. Xu, "Learning to model relationships for zero-shot video classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3476–3491, 2021. [Online]. Available: https://doi.org/10.1109/TPAMI.2020.2985708

[41] C. Ren, Y. H. Liu, X. Zhang, and K. Huang, "Multi-source unsupervised domain adaptation via pseudo target domain," *IEEE Trans. Image Process.*, vol. 31, pp. 2122–2135, 2022.

[42] M. Ning, D. Lu, Y. Xie, D. Chen, D. Wei, Y. Zheng, Y. Tian, S. Yan, and L. Yuan, "Madav2: Advanced multi-anchor based active domain adaptation segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13 553–13 566, 2023.

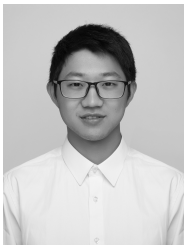
[43] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, "Domain-adversarial training of neural networks," in *Domain Adaptation in Computer Vision Applications*, ser. Advances in Computer Vision and Pattern Recognition, G. Csurka, Ed. Springer, 2017, pp. 189–209. [Online]. Available: https://doi.org/10.1007/978-3-319-58347-1_10

[44] Y. Cao, Y. Ma, M. Zhou, C. Liu, H. Xie, T. Ge, and Y. Jiang, "Geometry aligned variational transformer for image-conditioned layout generation," in *ACM Multimedia*. ACM, 2022, pp. 1561–1571.

[45] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *ECCV (5)*, ser. Lecture Notes in Computer Science, vol. 8693. Springer, 2014, pp. 740–755.

[46] B. Deka, Z. Huang, C. Franzen, J. Hibschman, D. Afergan, Y. Li, J. Nichols, and R. Kumar, "Rico: A mobile app dataset for building data-driven design applications," in *UIST*. ACM, 2017, pp. 845–854.

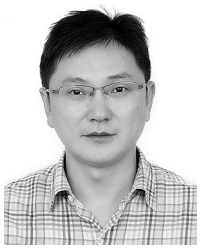
[47] T. F. Liu, M. Craft, J. Situ, E. Yumer, R. Mech, and R. Kumar, "Learning design semantics for mobile apps," in *UIST*. ACM, 2018, pp. 569–579.
[48] X. Zhong, J. Tang, and A. Jimeno-Yepes, "Publaynet: Largest dataset ever for document layout analysis," in *ICDAR*, 2019, pp. 1015–1022.
[49] B. Wang, Q. Chen, M. Zhou, Z. Zhang, X. Jin, and K. Gai, "Progressive feature polishing network for salient object detection," in *AAAI*. AAAI Press, 2020, pp. 12 128–12 135.
[50] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*. IEEE Computer Society, 2016, pp. 2818–2826.
[51] I. J. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," *CoRR*, vol. abs/1701.00160, 2017.
[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*. IEEE Computer Society, 2016, pp. 770–778.
[53] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *CVPR*. IEEE Computer Society, 2017, pp. 936–944.
[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.
[55] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 936–944.
[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
[57] 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, "Learning transferable visual models from natural language supervision," in *ICML*, vol. 139, 2021, pp. 8748–8763.
[58] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *ICCV*. IEEE, 2021, pp. 387–396.
[59] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu, "Layoutgan: Synthesizing graphic layouts with vector-wireframe adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2388–2399, 2021.
[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR, San Diego, CA, USA, May 7-9, 2015*, Y. Bengio and Y. LeCun, Eds., 2015.
[61] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.



Chenchen Xu received the B.Sc. and M.Sc. degrees from Anhui Normal University, Wuhu, China, in 2016 and 2020, respectively. He received the Ph.D. degree from the State Key Lab of CAD & CG, Zhejiang University, Hangzhou, China, in 2024, under the supervision of Prof. Weiwei Xu. He is currently a researcher at Anhui Normal University and The Chinese University of Hong Kong. His research interests include image processing and machine learning, with a focus on deep learning, graphic layout generation, and image matting.

Min Zhou received the B.S. and M.S. degrees from Beihang University, Beijing, China, in 2016 and 2019, respectively. She is currently a researcher in Alibaba Group. Her research interests include computer vision and deep learning.

Tiezheng Ge received his B.S. and Ph.D. degree from University of Science and Technology of China in 2009 and 2014 respectively. After that, he joined Alibaba Group. Now, he serves as a Staff Algorithm Engineer in Alimama(the Advertising Department of Alibaba), leading a research group of intellegent ad creative designing. His recent research interest includes the smart generation of image/video/text for online e-Commercial product, and relevant technics such as motion transfer, image matting, image/video caption, and image inpainting.



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.