Unsupervised Domain Adaption with Pixel-level Discriminator for Image-aware Layout Generation

Chenchen Xu[1,2*]   Min Zhou[2]   Tiezheng Ge[2]   Yuning Jiang[2]
Weiwei Xu[1†]
[1]State Key Lab of CAD&CG, Zhejiang University   [2]Alibaba Group
xuchenchen@zju.edu.cn, {yunqi.zm, tiezheng.gtz, mengzhu.jyn}@alibaba-inc.com, xww@cad.zju.edu.cn

Abstract

Layout is essential for graphic design and poster generation. Recently, applying deep learning models to generate layouts has attracted increasing attention. This paper focuses on using the GAN-based model conditioned on image contents to generate advertising poster graphic layouts, which requires an advertising poster layout dataset with paired product images and graphic layouts. However, the paired images and layouts in the existing dataset are collected by inpainting and annotating posters, respectively. There exists a domain gap between inpainted posters (source domain data) and clean product images (target domain data). Therefore, this paper combines unsupervised domain adaption techniques to design a GAN with a novel pixel-level discriminator (PD), called PDA-GAN, to generate graphic layouts according to image contents. The PD is connected to the shallow level feature map and computes the GAN loss for each input-image pixel. Both quantitative and qualitative evaluations demonstrate that PDA-GAN can achieve state-of-the-art performances and generate high-quality image-aware graphic layouts for advertising posters.

1. Introduction

Graphic layout is essential to the design of posters, magazines, comics, and webpages. Recently, generative adversarial network (GAN) has been applied to synthesize graphic layouts through modeling the geometric relationships of different types of 2D elements, for instance, text and logo bounding boxes [10, 19]. Fine-grained controls over the layout generation process can be realized using Conditional GANs, and the conditions might include image contents and the attributes of graphic elements, e.g. category, area, and aspect ratio [20, 35]. Especially, image

---

*Work done during an internship at Alibaba Group
†Corresponding author

Figure 1. **Examples of image-conditioned advertising posters graphic layouts generation.** Our model generates graphic layouts (middle) with multiple elements conditioned on product images (left). The designer or even automatic rendering programs can utilize graphic layouts to render advertising posters (right).

contents play an important role in generating image-aware graphic layouts of posters and magazines [34, 35].

This paper focuses on studying the deep-model based image-aware graphic layout method for advertising poster design, where the graphic layout is defined to be a set of elements with their classes and bounding boxes as in [35]. As shown in Fig. 1, the graphic layout for advertising poster design in our work refers to arranging four classes of elements, such as logos, texts, underlays, and other elements for embellishment, at the appropriate position according to product images. Therefore, its kernel is to model the relationship between the image contents and layout elements [4,35] such that the neural network can learn how to produce the aesthetic arrangement of elements around the product image. It can be defined as the direct set prediction problem in [5].

Constructing a high-quality layout dataset for the training of image-ware graphic layout methods is labor intensive, since it requires professional stylists to design the arrangement of elements to form the paired product image and layout data items. For the purpose of reducing workload, zhou et.al. [35] propose to collect designed poster images to construct a dataset with required paired data. Hence, the graphic elements imposed on the poster image are removed through image inpainting [28], and annotated with their geometric arrangements in the posters, which results in state-of-the-art CGL-Dataset with 54,546 paired data items. While CGL-Dataset is substantially beneficial to the training of image-ware networks, there exists a domain gap between product image and its inpainted version. The CGL-GAN in [35] tries to narrow this domain gap by utilizing Gaussian blur such that the network can take a clean product image as input for synthesizing a high-quality graphic layout. However, it is possible that the blurred images lose the delicate color and texture details of products, leading to unpleasing occlusion or placement of graphic elements.

This paper proposes to leverage unsupervised domain adaption technique to bridge the domain gap between clean product images and inpainted images in CGL-Dataset to significantly improve the quality of generated graphic layouts. Treating the inpainted poster images without graphic elements as the source domain, our method aims to seek for the alignment of the feature space of source domain and the feature space of clean product images in the target domain. To this end, we design a GAN with a pixel-level domain adaption discriminator, abbreviated as PDA-GAN, to achieve more fine-grained control over feature space alignment. It is inspired by PatchGAN [13], but non-trivially adapts to pixel-level in our task. First, the pixel-level discriminator (PD) designed for domain adaption can avoid the Gaussian blurring step in [35], which is helpful for the network to model the details of the product image. Second, the pixel-level discriminator is connected to the shallow level feature map, since the inpainted area is usually small relative to the whole image and will be difficult to discern at deep levels with large receptive field. Finally, the PD is con-

structed by three convolutional layers only, and its number of network parameters is less than 2% of the discriminator parameters in CGL-GAN. This design reduces the memory and computational cost of the PD.

We collect 120,000 target domain images during the training of PDA-GAN. Experimental results show that PDA-GAN achieves state-of-the-art (SOTA) performance according to composition-relevant metrics. It outperforms CGL-GAN on CGL-dataset and achieves relative improvement over background complexity, occlusion subject degree, and occlusion product degree metrics by 6.21%, 17.5%, and 14.5% relatively, leading to significantly improved visual quality of synthesized graphic layouts in many cases. In summary, this paper comprises the following contributions:

- We design a GAN with a novel pixel-level discriminator working on shallow level features to bridge the domain gap that exists between training images in CGL-Dataset and clean product images.

- Both quantitative and qualitative evaluations demonstrate that PDA-GAN can achieve SOTA performance and is able to generate high-quality image-aware graphic layouts for advertising posters.

2. Related Work

Image-agnostic layout generation. Early works [3, 14, 17, 24] often utilize templates and heuristic rules to generate layouts. LayoutGAN [19] is the first method to apply generative networks (in particular GAN) to synthesize layouts and use self-attention to build the element relationship. LayoutVAE [15] and LayoutVTN [1] follow and apply VAE and autoregressive methods. Meanwhile, some conditional methods have been proposed to guide the layout generation process [11, 16, 18, 20, 31]. The constraints are in various forms, such as scene graphs, element attributes, and partial layouts. However, in a nutshell, these methods mainly focus on modeling the internal relationship between graphic elements, and rarely consider the relationship between layouts and images.

Image-aware layout generation. In layout generation for magazine pages, ContentGAN [34] first proposes to model the relationship not only between layout elements but also between layouts and images. However, the high-quality training data is relatively rare, since it requires professional stylists to design layouts for obtaining paired clean images and layouts. ContentGAN uses white patches to mask the graphic elements on magazine pages, and replaces the clean images with the processed pages for training. For the same problem, in poster layout generation, CGL-GAN [35] leverages inpainting to erase the graphic elements on posters, and
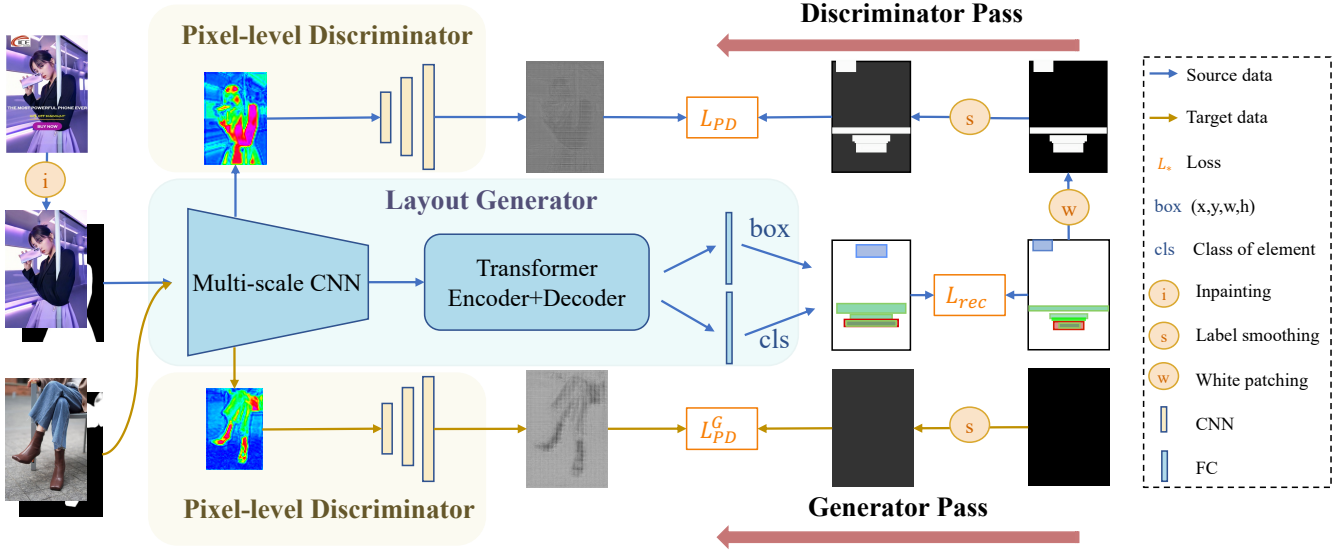
Figure 2. **The architecture of our network.** Annotated posters (source domain data) must be inpainted before input to the model. The model has both reconstruction and GAN loss when training with source domain data, while only has a GAN loss is used when training with target domain data. Please refer to Sec. 3 for the definition of each loss term: L_{PD}, L_{PD}^G, and L_{rec}. During the discriminator or generator pass, both inpainted and clean images are fed into the discriminator.

subsequently applies Gaussian blur on the whole poster to eliminate the inpainting artifacts. The blur strategy effectively narrows the domain gap between inpainted images and clean images, but it may damage the delicate color and texture details of images and leads to unpleasing occlusion or element placement. In this paper, we find that a pixel-level discriminator for domain adaption can achieve the same goal and avoid the negative effects of blur.

Unsupervised domain adaptation. Unsupervised domain adaptation [7] 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 [2, 7, 22, 23, 25, 27, 32, 33] have been applied for object recognition and detection. Among these methods, [2, 7, 25, 27] leverage adversarial domain adaptation approach [8]. A domain discriminator is employed and outputs a probability value indicating the domain of input data. In this way, the generator can extract domain-invariant features and eliminate 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 to effectively solve this.

3. Our Model

Our model is a generative adversarial network to learn domain-invariant features with the pixel-level discriminator

to minimize the cross-domain discrepancy. As shown in Fig. 2, our network mainly has two sub-networks: the layout generator network that takes the image and its saliency map 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 our network architecture and the training loss functions for the pixel-level discriminator and the layout generator network respectively.

3.1. Network Architecture

The architecture of the layout generator network is the same with the generator network in [35], but the user-constraints are ignored. Its design follows the principle of DETR [5], which has three modules: a multi-scale convolutional neural network (CNN) used to extract image features [12, 21], a transformer encoder-decoder that accepts layout element queries as input to model the relationship among layout elements and the product image [30], and two fully connected layers to predict the element class and its bounding box using the element feature output by the transformer decoder.

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 multi-scale CNN. The transposed convolutional layers can up-sample the feature map, and we also allow to resize the final result to exactly match the dimension of the input image to facilitate the computation of discriminator training loss, which will be elaborated in the next section.

Model	$R_{com}\downarrow$	$R_{shm}\downarrow$	$R_{sub}\downarrow$	$R_{ove}\downarrow$	$R_{und}\uparrow$	$R_{ali}\downarrow$	$R_{occ}\uparrow$
ContentGAN [34]	45.59	17.08	1.143	0.0397	0.8626	**0.0071**	93.4
CGL-GAN [35]	<u>35.77</u>	<u>15.47</u>	<u>0.805</u>	**0.0233**	<u>0.9359</u>	<u>0.0098</u>	<u>99.6</u>
PDA-GAN(Ours)	**33.55**	**12.77**	**0.688**	<u>0.0290</u>	**0.9481**	0.0105	**99.7**

Table 1. **Comparison with content-aware methods.** Bold and underlined numbers denote the best and second best respectively.

3.2. Pixel-level Discriminator Training

The design of pixel-level discriminator is based on the observation that the domain gap between inpainted images and clean product images mainly exists at pixels synthesized by inpainting process. Therefore, during the discriminator or generator pass in Fig. 2, both inpainted and clean images are fed into the discriminator. When updating the discriminator, we encourage the discriminator to detect the inpainted pixels for inpainted images 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 the inpainted images, the discriminator's ability to detect inpainted pixels should be weakened fast. In this way, when the training converges, the feature space of source and target domain images should be aligned.

To calculate the loss L_{PD} for each input-image pixel, we utilized the white-patch map to distinguish whether the input-image pixel is inpainted, where the pixel in white patch map is set to 1 if the corresponding pixel in the input image is processed by the inpainting, otherwise 0. Correspondingly, the pixel values of white-patch map for the clean images in target domain are all 0.

When updating 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. We can get as:

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

where the N_p means the number of white-patch map pixels, and $\mathbf{P}_i$ indicates the predicted or ground-truth map for i_{th} image. The superscript of $\mathbf{P}_i$ indicates it is from source by s or target by t, from prediction by o or ground truth by w. The two coefficients, α and β, are used to balance between the source and target domain white-patch map. Since the area of the inpainted pixels in the white-patch map are usually small, we set the value of α to 2 and β to 1.

We utilize one-side label smoothing [9, 29] to improve the generalization ability of the trained model. Since the inpainted areas occupy a small proportion of the input image, we only do label smoothing for pixels not in the inpainted area (those pixels with value 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.

3.3. Layout Generator Network Training

When updating the generator network in the GAN training, we expect 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 pixels with value 1. Thus, we have:

$$L_{PD}^G = \frac{1}{N_p}\sum_{i=1}^{N_p}(|\hat{\mathbf{P}}_i^s - \mathbf{P}_i^{s,o}| * \alpha \\ + |\mathbf{P}_i^{t,w} - \mathbf{P}_i^{t,o}| * \beta),$$ (2)

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

$$L_G = L_{rec} + \gamma * L_{PD}^G,$$ (3)

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].

4. Experiments

In this section, we mainly compare our model with SOTA layout generation methods and its ablation studies. More additional experimental analyses and designed advertising posters using generated layouts can be found in the supplementary materials.

4.1. Implementation Details

We implement our PDA-GAN in PyTorch and use Adam optimizer for training the model. The initial learning rates are 10^{-5} for the generator backbone and 10^{-4} for the remaining part of this model. The model is trained for 300
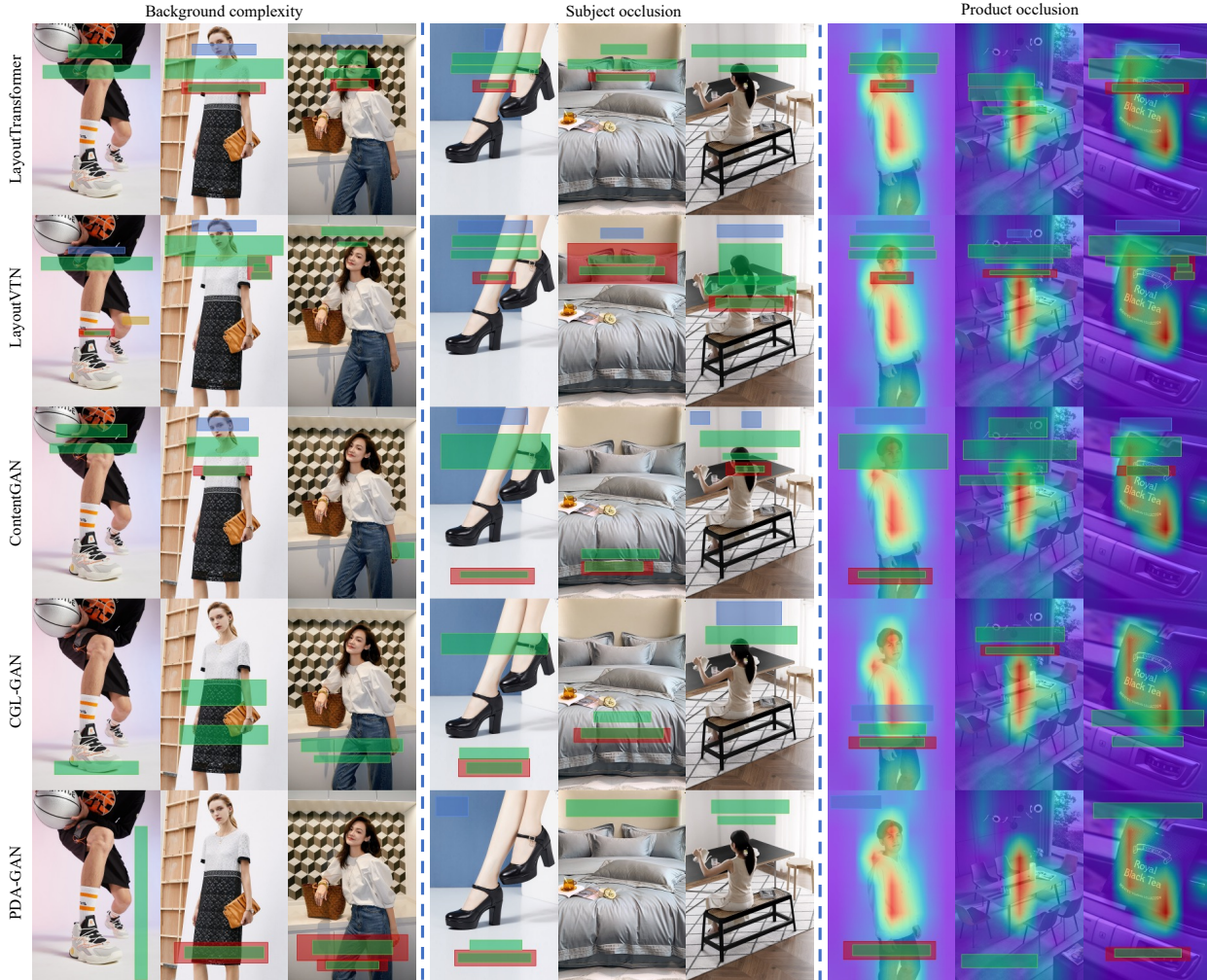
Figure 3. **Qualitative evaluation for different models.** Layouts in a column are conditioned with the same image. And those in a row are from the same model. This figure qualitatively compares and analyzes different models from three aspects: text element background complexity, overlapping subject and overlapping product attention map.

Model	R_{com} ↓	R_{shm} ↓	R_{sub} ↓	R_{ove} ↓	R_{und} ↑	R_{ali} ↓
LT	40.92	21.08	1.310	0.0156	0.9516	0.0049
VTN	41.77	22.21	1.323	**0.0130**	**0.9698**	**0.0047**
Ours	**33.55**	**12.77**	**0.688**	0.0290	0.9481	0.0105

Table 2. **Comparison with content-agnostic methods.** LT and VTN represent LayoutTransformer and LayoutVTN, repectively.

epochs with a batch size of 128, and all learning rates are reduced by a factor of 10 after 200 epochs. To make the fair experimental comparisons, we follow CGL-GAN [35] to re-size the inpainted posters and product images to 240×350 as inputs of our PDA-GAN. The total training time is about 8 hours using 16 NVIDIA V100 GPUs.

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 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 a comparison, we refer to it as Data II. In the following, if not clearly mentioned, our model is trained with Data I.

4.2. Metrics

For quantitative evaluations, we follow [35] to divide layout metrics into the composition-relevant metrics and the graphic metrics. The composition-relevant metrics include
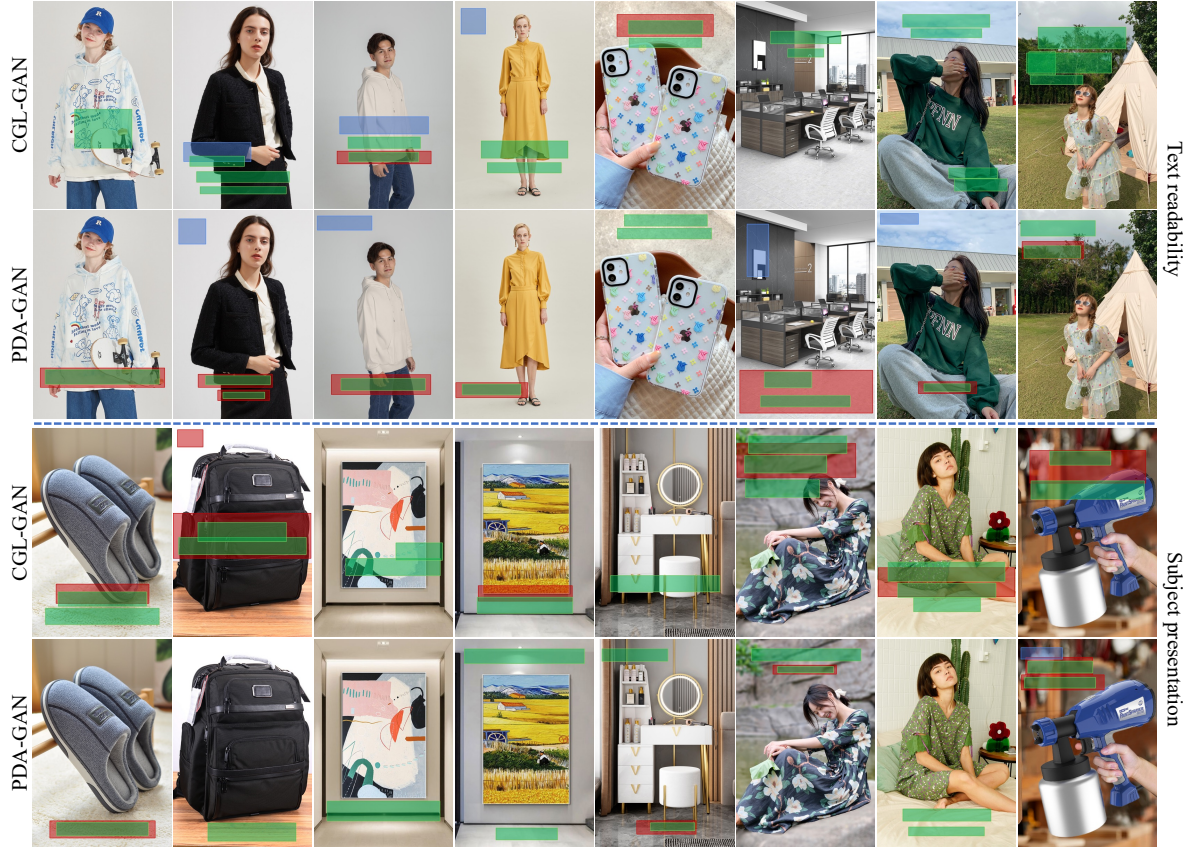
Figure 4. More qualitative comparisons with CGL-GAN.

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 (Ours)	✓			**33.55**	**12.77**	**0.688**	0.0290	**0.9481**	0.0105	**99.7**

Table 3. **Comprehensive comparison between CGL-GAN and PDA-GAN (Ours).** 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.

R_{com}, and R_{shm}, R_{sub}, which measure background complexity, occlusion subject degree, and occlusion product degree respectively; while the graphic metrics include R_{ove}, R_{und}, and R_{ali}, which measure layout overlap, underlay overlap, and layout alignment degree respectively. When the R_{ove} value of a model exceeds 0.05, the generated element bounding boxes will overlap each other severely, resulting in useless layouts. This means that the high value of R_{ove} indicates a failure in the layout generation for most images. Moreover, we use metric R_{occ} to represent the ratio of non-empty layouts predicted by models. We will use all the above metrics to compare each group's experiments to verify the effectiveness of our model. The formal

definitions of these metrics and corresponding examples of graphic layouts are shown in the supplementary material.

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

Layout generation with image contents. We first conduct experiments to compare our method with Content-GAN and CGL-GAN that can generate image-aware layouts. The quantitative results can be seen from Tab. 1. Our model achieves the best results in most metrics, especially in the composition-relevant metrics since PDA-GAN preserves the image color and texture details. For instance, our PDA-GAN outperforms contentGAN and CGL-GAN by 26.4% and 6.21% respectively, with regard to back-

ground complexity R_{com}. As shown in the first column in Fig. 3, compared with these by contentGAN and CGL-GAN, bounding boxes of text element 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.

Comparing contentGAN and CGL-GAN, our PDA-GAN reduces the occlusion subject degree R_{shm} by 25.2% and 17.5% respectively. From the middle three columns of Fig. 3, for contentGAN or CGL-GAN, the presentation of the subject content information are 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 unpleasing, 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 occlusion product degree R_{sub} of PDA-GAN performance surpass contentGAN and CGL-GAN by 39.8% and 14.5% respectively. The three rightmost columns in Fig. 3 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 [6, 26]. 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 makes the hoodie information of the product can be fully displayed. The above quantitative and qualitative comparisons of models demonstrate that PDA-GAN improves the relationship modeling between image contents and graphic layouts.

Layout generation without image contents. We also compare with image-agnostic methods of LayoutTransformer [11] and LayoutVTN [1]. As shown in Tab. 2, these image-agnostic methods perform pretty well on graphic metrics. However, in term of composition-relevant metrics, our model is much better. In detail, our PDA-GAN beyonds LayoutTransformer and LayoutVTN by 18.0% and 19.7% respectively with regard to R_{com}. That is because these image-agnostic methods only care about the relationship between elements while do not account for image contents. These image-agnostic methods are prone to generate bounding boxes of text elements in the area with complex backgrounds(as shown in the first two rows and the left-

most three columns of Fig. 3), which will reduce the readability of the text information. Furthermore, compared with LayoutTransformer and LayoutVTN, R_{shm} of PDA-GAN is reduced by 39.4% and 42.5%, and R_{sub} is reduced by 47.5% and 48.0%. The rightmost six columns in Fig. 3 show image-agnostic methods generate layout bounding boxes that randomly occlusion on the subject and product areas. These bounding boxes of layout elements will diminish the content and information presentation of the subject and product.

More comparisons with CGL-GAN. As shown in the first and the last row of Tab. 3, PDA-GAN outperforms CGL-GAN in all metrics with the same configuration. PDA-GAN differs from CGL-GAN that it uses PD to replace the discriminator in CGL-GAN. The number of the PD parameters is 332,545, less than 2% of the discriminator (22,575,841) in CGL-GAN, which significantly reduces the memory and computation cost of PDA-GAN. The second row of Tab. 3 shows that the model training on Data I with Gaussian blur performs poorly on R_{ove}, which causes most bounding boxes to overlap each other. Intuitively, Gaussian blur can narrow the domain gap, but it will also cause the image color and texture details lost. From the last two rows of Tab. 3, PDA-GAN without Gaussian blur is far better than CGL-GAN with Gaussian blur on composition-relevant metrics. As illustrated in the the sixth and eighth columns of the first two rows in Fig. 4, compared with CGL-GAN, PDA-GAN generates text bounding boxes with the simpler background. It is interesting to observe from the first two rows of Fig. 4 that when PDA-GAN generates box among complexity background, it tends to additionally generate an underlay bounding box which covers the complex background to ensure the 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 better express the information of subjects and layout elements.

Both 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. Refer to the supplementary for more details.

4.4. Ablations

Effects of pixel-level Discriminator. We first compare our PD with a global discriminator that only predicts one real or fake probability as in classical GAN. The abbreviation DA in Tab. 4 indicates the global discriminator strategy. When the weight of DA loss (γ in Eq. (3)) 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. 4, our PD outperforms DA on all metrics.

Second, we compare the PD with the strategy in Patch-

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

Table 4. **Ablation study with discriminator level.** DA indicates the global discriminator strategy in classical GAN-based methods, which outputs one probability value for real or fake for an image. The PDA output is a pixel-level map, and its dimension is same with the input image. W refers to the weight of DA (or PDA) module loss in the training process. Please refer to Tab. 3 for the explanation of the symbol "-".

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

Table 5. **Quantitative ablation study PatchGAN-based methods.** Patch size means the size of the map output by the discriminator. The input image height and width are 320 and 240 respectively. We train these models with γ in Eq. (3) equal to 6. Please refer to Tab. 3 for the explanation of the symbol "-".

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 6. **Quantitative ablation study on different level feature maps for pixel-level discriminator.**

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

Table 7. **Ablation study on different label smoothing choice.** 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.

Effects of label smoothing. Tab. 7 shows that the model with one-target label smoothing performs better in all metrics than without label smoothing. In addition, the effects of two-side or one-source label smoothing are not as good as one-target label smoothing on average. For the ground truth map input to the discriminator, the two-side label smoothing means we set 0 to 0.2 and 1 to 0.8, and one-source label smoothing means we only set 1 to 0.8.

5. Conclusion

In this paper, we study the domain gap problem between clean product images and inpainted images in CGL-Dataset for generating poster layouts. To solve this problem, we propose to leverage the unsupervised domain adaptation technique and design a pixel-level discriminator. This design of discriminator can not only finely align image features of these two domains, but also avoid the Gaussian blurring step in the previous work (CGL-GAN), which brings benefits to modeling the relationship between image details and layouts. Both quantitative and qualitative evaluations demonstrate that our method can achieve SOTA performance and generate high-quality image-aware graphic layouts for posters. In the future, we may investigate how to better interact with user constraints, e.g. categories and coordinates of elements, and enhance the layout generation diversity.

Acknowledgements

GAN [13]. The scores of quantitative metrics listed in Tab. 5 also verify the advantage of PD. The patch size in this table means the dimension of the output map, which will be compared with correspondingly resized ground-truth white patch map during training. These experiments show that, since the discrepancy by inpainting exists between pixels, the model might be required to eliminate the domain gap at the pixel level. Moreover, the pixel-level strategy can be considered as the most fine-grained patch level strategy.

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 how the PD works if the deep level feature, i.e. the feature from fourth residual block, and the fused feature in multi-scale CNN [35], i.e. the fusion of first to fourth residual block feature map, are used in the PD. As shown in Tab. 6, discriminating with shallow feature map in PDA-GAN can achieve better results in both composition-relevant and graphic metrics on average. Again, this experiment verifies the advantage of our design of PD. Intuitively, bridging the domain gap at early stage of the network might be beneficial to the subsequent model processing.

References

[1] Diego Martín Arroyo, Janis Postels, and Federico Tombari. Variational transformer networks for layout generation. In *CVPR*, pages 13642–13652. Computer Vision Foundation / IEEE, 2021. 2, 7

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, pages 95–104. IEEE Computer Society, 2017. 3

[3] Ying Cao, Antoni B. Chan, and Rynson W. H. Lau. Automatic stylistic manga layout. *ACM Trans. Graph.*, 31(6):141:1–141:10, 2012. 2

[4] Yunning Cao, Ye Ma, Min Zhou, Chuanbin Liu, Hongtao Xie, Tiezheng Ge, and Yuning Jiang. Geometry aligned variational transformer for image-conditioned layout generation. In *ACM Multimedia*, pages 1561–1571. ACM, 2022. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV (1)*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020. 2, 3, 4

[6] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *ICCV*, pages 387–396. IEEE, 2021. 7

[7] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R. Arabnia. A brief review of domain adaptation. *CoRR*, abs/2010.03978, 2020. 3

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. In Gabriela Csurka, editor, *Domain Adaptation in Computer Vision Applications*, Advances in Computer Vision and Pattern Recognition, pages 189–209. Springer, 2017. 3

[9] Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017. 4

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 1

[11] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *ICCV*, pages 984–994. IEEE, 2021. 2, 7

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016. 3

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 5967–5976. IEEE Computer Society, 2017. 2, 8

[14] Charles E. Jacobs, Wilmot Li, Evan Schrier, David Bargeron, and David Salesin. Adaptive grid-based document layout. *ACM Trans. Graph.*, 22(3):838–847, 2003. 2

[15] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In *ICCV*, pages 9894–9903. IEEE, 2019. 2

[16] Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Constrained graphic layout generation via latent optimization. In *ACM Multimedia*, pages 88–96. ACM, 2021. 2

[17] Ranjitha Kumar, Jerry O. Talton, Salman Ahmad, and Scott R. Klemmer. Bricolage: example-based retargeting for web design. In *CHI*, pages 2197–2206. ACM, 2011. 2

[18] Hsin-Ying Lee, Lu Jiang, Irfan Essa, Phuong B. Le, Haifeng Gong, Ming-Hsuan Yang, and Weilong Yang. Neural design network: Graphic layout generation with constraints. In *ECCV (3)*, volume 12348 of *Lecture Notes in Computer Science*, pages 491–506. Springer, 2020. 2

[19] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. Layoutgan: Generating graphic layouts with wireframe discriminators. In *ICLR (Poster)*. OpenReview.net, 2019. 1, 2

[20] Jianan Li, Jimei Yang, Jianming Zhang, Chang Liu, Christina Wang, and Tingfa Xu. Attribute-conditioned layout GAN for automatic graphic design. *IEEE Trans. Vis. Comput. Graph.*, 27(10):4039–4048, 2021. 1, 2

[21] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944. IEEE Computer Society, 2017. 3

[22] Debjeet Majumdar and Vinay P. Namboodiri. Unsupervised domain adaptation of deep object detectors. In *ESANN*, 2018. 3

[23] Sushruth Nagesh, Shreyas Rajesh, Asfiya Baig, and Savitha Srinivasan. Domain adaptation for object detection using SE adaptors and center loss. *CoRR*, abs/2205.12923, 2022. 3

[24] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. Learning layouts for single-pagegraphic designs. *IEEE Trans. Vis. Comput. Graph.*, 20(8):1200–1213, 2014. 2

[25] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, pages 3934–3941. AAAI Press, 2018. 3

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 7

[27] Chuan-Xian Ren, Yong Hui Liu, Xiwen Zhang, and Ke-Kun Huang. Multi-source unsupervised domain adaptation via pseudo target domain. *IEEE Trans. Image Process.*, 31:2122–2135, 2022. 3

[28] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, pages 3172–3182. IEEE, 2022. 2

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826. IEEE Computer Society, 2016. 4

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 3

[31] Cheng-Fu Yang, Wan-Cyuan Fan, Fu-En Yang, and Yu-Chiang Frank Wang. Layouttransformer: Scene layout generation with conceptual and spatial diversity. In *CVPR*, pages 3732–3741. Computer Vision Foundation / IEEE, 2021. 2

[32] Jingyi Zhang, Jiaxing Huang, Zichen Tian, and Shijian Lu. Spectral unsupervised domain adaptation for visual recognition. In *CVPR*, pages 9819–9830. IEEE, 2022. 3

[33] Youshan Zhang and Brian D. Davison. Domain adaptation for object recognition using subspace sampling demons. *Multim. Tools Appl.*, 80(15):23255–23274, 2021. 3

[34] Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson W. H. Lau. Content-aware generative modeling of graphic design layouts. *ACM Trans. Graph.*, 38(4):133:1–133:15, 2019. 1, 2, 4

[35] Min Zhou, Chenchen Xu, Ye Ma, Tiezheng Ge, Yuning Jiang, and Weiwei Xu. Composition-aware graphic layout GAN for visual-textual presentation designs. In *IJCAI*, pages 4995–5001. ijcai.org, 2022. 1, 2, 3, 4, 5, 8