Supplementary Material for LayoutDM

Due to limited space, our paper does not show all the comparison results. In this document, we will give more details about our experiments and present additional results.

1. Datasets

We summarize the statistics of the datasets used in our paper in Table 1. In all our experiments, we follow the following settings for training, validation and testing. For PublayNet, COCO, and TextLogo3K, we use 95% of the official training split for training, the rest for validation, and the official validation split for testing. For Rico and Magazine, we use 85% of the dataset for training, 5% for validation, and 10% for testing.

Dataset	# label types	Max. # elements	# train.	# val.	# test.
Rico	13	9	17,515	1,030	2061
PublayNet	5	9	160,549	8,450	4,226
Magazine	5	33	3,331	196	392
COCO	183	8	71,038	3,739	3,097
TextLogo3K	N/A	20	3,011	159	300

Table 1. Statistics of the datasets used in our experiments.

2. About Evaluation Metrics

Previous work often uses different metrics to evaluate the visual quality and realism of the generated layouts. Though some metrics have the same name (such as Alignment and Overlap), they are computed in different ways. In our paper, we use the metrics in [4] to evaluate our method, which is different from those in BLT [5]. Here, we detail how to compute these metrics.

2.1. Metrics in our paper

In our paper, we follow the guidance in [4] to evaluate our proposed method. Four evaluation metrics are used to evaluate the quality of the generated layouts: FID [4], Max. IoU, Alignment [8] and Overlap [8].

FID [4] measures the realism and accuracy of the generated layouts. To compute FID, we train a binary layout classifier to discriminate between real layouts and noise added layouts, and use the intermediate features of the network as the representative features of layouts. For a fair comparison, we use the pre-trained neural network in [4] to calculate the FID metrics.

Max. IoU [4] is defined between two collections of generated layouts and references. We compute the Max. IoU metric as in [4]. In our paper, the layout geometric sequence $\boldsymbol{g} = (g_1, \cdots, g_N)$ corresponds to the layout $B = \{b_i\}_{i=1}^N$ in [4], and the layout attributes sequence $\boldsymbol{f} = (f_1, \cdots, f_N)$ corresponds to the label set $l_i, i = 1, \cdots, N$ in [4].

Alignment [8] computes an alignment loss with the intuition that objects in graphic design are often aligned either by center or edge. Denote $\boldsymbol{\theta} = (x^L, y^T, x^C, y^C, x^R, y^B)$ as the top-left, center and bottom-right coordinates of the bounding box, we calculate the Alignment loss for each layout as follows:

$$L_{alg} = \frac{1}{N} \sum_{i=1}^{N} \min \left(\begin{array}{c} \boldsymbol{g}\left(\Delta x_i^L\right), \boldsymbol{g}\left(\Delta x_i^C\right), \boldsymbol{g}\left(\Delta x_i^R\right) \\ \boldsymbol{g}\left(\Delta y_i^T\right), \boldsymbol{g}\left(\Delta y_i^C\right), \boldsymbol{g}\left(\Delta y_i^B\right) \end{array} \right)$$

where $\boldsymbol{g}(x) = -\log(1-x)$, N is the number of elements in the layout, and $\Delta x_i^*(* = L, C, R)$ is computed as:

$$\Delta x_i^* = \min_{\forall j \neq i} \left| x_i^* - x_j^* \right|$$

$\Delta y_i^*(* = T, C, B)$ can be computed similarly. The final value is multiplied by 100× for visibility following [4].

Overlap [8] measures the total overlapping area between any pair of bounding boxes in a layout. We compute the Overlap metric for each layout as follows:

$$L_{over} = \frac{1}{N} \sum_{i=1}^{N} \sum_{\forall j \neq i} \frac{s_i \cap s_j}{s_i}$$

where $s_i \cap s_j$ denotes the overlapping area between element i and j. N is the number of elements in the layout. The final value is multiplied by 100× for visibility following [4].

2.2. Metrics in BLT

In this paper, we also follow the guidance of BLT [5] to evaluate our proposed method. The evaluation metrics used in BLT [5] are different from those in [4]: IoU, Alignment, and Overlap.

IoU [5] measures the intersection over the union beween the generated bounding boxes. Following [5], an improved perceptual IOU is used. It first projects the layouts as if they were images then computes the ovarlapped area divided by **the union area of all objects**. A toy sample images is presented in Fig. 1 [1]

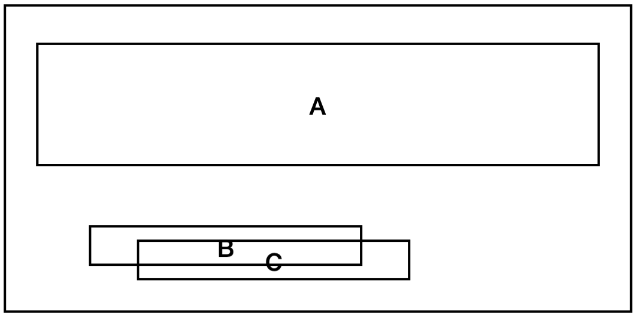

Figure 1. A toy layout sample for the IoU computation. The areas of objects A, B and C are 5, 1, 1, the overlapped area of B and C are 0.5. IoU $\frac{0.5}{6.5} = \frac{1}{13}$.

Alignment [6] measures the alignment among design elements. The average Alignment on a collection of layouts is computed using:

$$\frac{1}{N_D} \sum_d \sum_i \min_{j,i \neq j} \left\{ \min \left(l\left(c_i^d, c_j^d\right), m\left(c_i^d, c_j^d\right), r\left(c_i^d, c_j^d\right) \right\} \right)$$

where N_D is the number of generated layouts, c_k^d is the k-th component of the d-th layout. In addition, l, m, and r are alignment functions where the distances between the left, center, and right of elements are measured, respectively. [5] uses L1 distance as the distance measure but does not adopt the function $\boldsymbol{g}(x) = -\log(1-x)$ to process the distance. [5] also do not normalize the Alignment by the number of elements N or multiply the value by $100\times$. This leads to different results using the two different Alignments.

Overlap [7] is the percentage of total overlapping area among any two bounding boxes inside the whole page.

$$L_{over} = \sum_{i=1}^{N} \sum_{\forall j \neq i} \frac{s_i \cap s_j}{s_i}$$

[5] does not normalize the values with the number of elements N or multiply the value by $100\times$. This leads to different results using the two different Overlaps.

3. Comparison with BLT

BLT [5] is another method for conditional layout generation, which is recently published in ECCV2022. Due to limited space, we only compare our LayoutDM with BLT on the PublayNet dataset. Here, we show more comparison results on more datasets using the metrics of BLT under the setting of conditional layout generation. The SOTA methods being compared include LayoutVAE [3], LayoutTransformer [2], VTN [1], NDN [6], LayoutGAN++ [4] and BLT [5]. Since most SOTA methods (except LayoutGAN++) have no available public implements, we directly cite the experimental results in [5]. All the experiments run five times, and the mean and standard deviation over five trials are reported. The full comparison results are shown in Tab. 2. As we can see, our LayoutDM significantly outperforms SOTA methods on PublayNet, Rico, and Maganize.

4. Why LayoutDM has lower FID value than "Real Data"?

The reason lies in the definition of FID metric. According to [4], the FID value measures the distribution distance between the layouts generated by the models and the real layouts in *test set*. For "real data", the FID value is computed using the real layouts in *validation set* and those in *test set*. Since we generate the layouts conditioned on the attributes of layouts in test set, the distribution of generated layouts should be more similar to that of the test set than validation set. So, our method has a lower FID value than "real data".

Moreover, to verify our analysis, we generate layouts conditioned on the layout attributes in the validation set, and compute the FID value of our method (denoted as "Gen./Val. Set" in the following table). The result is shown in Tab. 4. In this case, we can observe that the FID value of our method is close to that of "Real data", which is consistent with our analysis.

5. Ablation study on positional encoding.

Qualitative results. In this paper, we don't consider the order of designed elements on a canvas, so our LayoutDM omits the positional encoding in its transformer component. To evaluate the impact of positional encoding, we train a LayoutDM model with positional encoding as a reference (denoted as LayoutDM/PE). We randomly shuffle the order of elements in the input sequence in all three data sets, and evaluate the LayoutDM and LayoutDM/PE in the shuffled sequence.

The results are shown in Fig. 2. As we can see, after removing the positional encoding, the model is no longer aware of the positional information in the input sequence, and thus can still generate reasonable results conditioned on the shuffled input sequence. If we use positional encoding in our model, the model (LayoutDM/PE) will interpret the original sequence and the shuffled sequence as completely

[1] Metric description and sample image are cited from the appendix of [5]

Dataset	PublayNet			Rico			Magazine
Model	IoU↓	Overlap↓	Alignment↓	IoU↓	Overlap↓	Alignment↓	IoU↓
L-VAE [3]	0.45±1.3%	0.15±0.9%	0.37±0.7%	0.41±1.5%	0.39±2.3%	0.38±1.9%	\
NDN [6]	0.34±1.8%	0.12±0.8%	0.39±0.4%	0.37±1.7%	0.36±1.9%	0.41±1.6%	\
VTN [1]	0.21±0.6%	0.06±0.2%	0.33±0.4%	0.30±0.1%	0.30±0.3%	0.32±0.9%	0.18±1.8%
Trans. [2]	0.19±0.3%	0.06±0.3%	0.33±0.3%	0.31±0.2%	0.33±0.8%	0.30±0.8%	0.20±0.8%
BLT [5]	0.19±0.2%	0.04±0.1%	0.25±0.7%	0.30±0.4%	0.23±0.2%	**0.20**±1.1%	0.18±0.6%
LayoutGAN++ [4]	0.084±1.1%	0.05±0.3%	0.24±0.5%	**0.22**±1.2%	0.18±0.85%	0.25±0.6%	0.061±1.7%
LayoutDM(Ours)	**0.0053**±0.5%	**0.01**±0.1%	**0.22**±1.2%	**0.22**±0.7%	**0.17**±0.7%	0.24±1.2%	**0.055**±0.9%
Real Data	0.00097	0.0006	0.283	0.19	0.167	0.277	0.054

Table 2. Quantitative comparison using the evaluation metrics in BLT. "Trans." and "L-VAE" denote "LayoutTransformer" and "LayoutVAE". The metrics computed with the test datasets are shown as Real Data.

Dataset	Rico			
Model	FID↓	Max. IoU↑	Alignment↓	Overlap↓
LayoutDM/PE (shuffled)	14.97±0.48	0.35±0.00	0.42±0.05	55.65±0.31
LayoutDM/PE (original)	3.26±0.08	0.48±0.01	0.32±0.05	57.45±0.27
LayoutDM (shuffled)	4.04±0.10	0.46±0.00	0.29±0.02	58.32±0.23
LayoutDM (original)	3.95±0.09	0.46±0.00	0.28±0.04	58.59±0.40
Dataset	PublayNet			
Model	FID↓	Max. IoU↑	Alignment↓	Overlap↓
LayoutDM/PE (shuffled)	29.83±0.54	0.34±0.01	0.28±0.01	16.03±0.27
LayoutDM/PE (original)	3.96±0.08	0.44±0.00	0.14±0.00	3.93±0.09
LayoutDM (shuffled)	4.15±0.10	0.44±0.00	0.14±0.01	4.23±0.07
LayoutDM (original)	4.17±0.12	0.44±0.00	0.14±0.01	4.13±0.05
Dataset	Magazine			
Model	FID↓	Max. IoU↑	Alignment↓	Overlap↓
LayoutDM/PE (shuffled)	19.23±0.54	0.24±0.00	0.89±0.01	54.59±1.35
LayoutDM/PE (original)	9.44±0.13	0.29±0.00	0.84±0.02	35.64±0.68
LayoutDM (shuffled)	10.05±0.19	0.28±0.00	0.78±0.04	33.76±0.81
LayoutDM (original)	9.75±0.40	0.29±0.00	0.79±0.02	32.74±0.58

Table 3. Quantitative results on the effect of element order.

	Gen. / Test set	**Gen. / Val. set**	Real data
FID↓	4.04±0.08	**10.86±0.04**	9.54

Table 4. FID comparison on PublayNet. Gen. denotes "Generated" and Val. denotes "Validation".

different inputs and thus produces results of bad quality.

Quantitative results. Table 3 shows the quantitative results. As one can see, shuffling the order of elements does not affect the model without positional encoding, while the performance of the model with positional encoding is significantly affected. This proves that the performance of our model is independent of the order of the elements in the input sequence.

6. More Qualitative Results

This section presents more qualitative results. We show the comparison results of three models in Fig. 3, namely our implemented conditional VTN, LayoutGAN++ and LayoutDM. As one can see, our model has higher quality in terms of alignment and overlap than the other two methods. The layouts generated by our LayoutDM are very similar to the real data.

7. More Qualitative Comparisons on Diversity

We provide more quantitative comparisons on diversity in Fig. 4. It can be observed that compared to the other two methods, our model has better diversity while maintaining high quality.

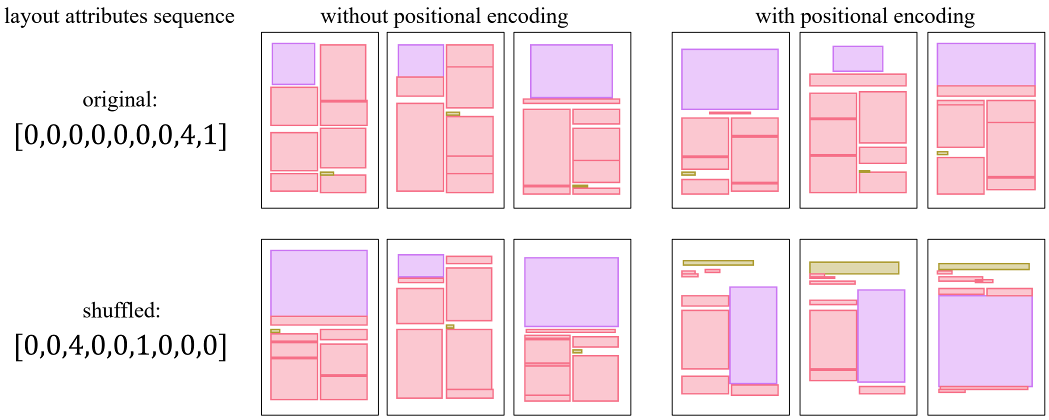

Figure 2. Effect of element order of input sequence. We use index "0" , "1" and "4" to represent element category label "Text", "Title" and "Figure".

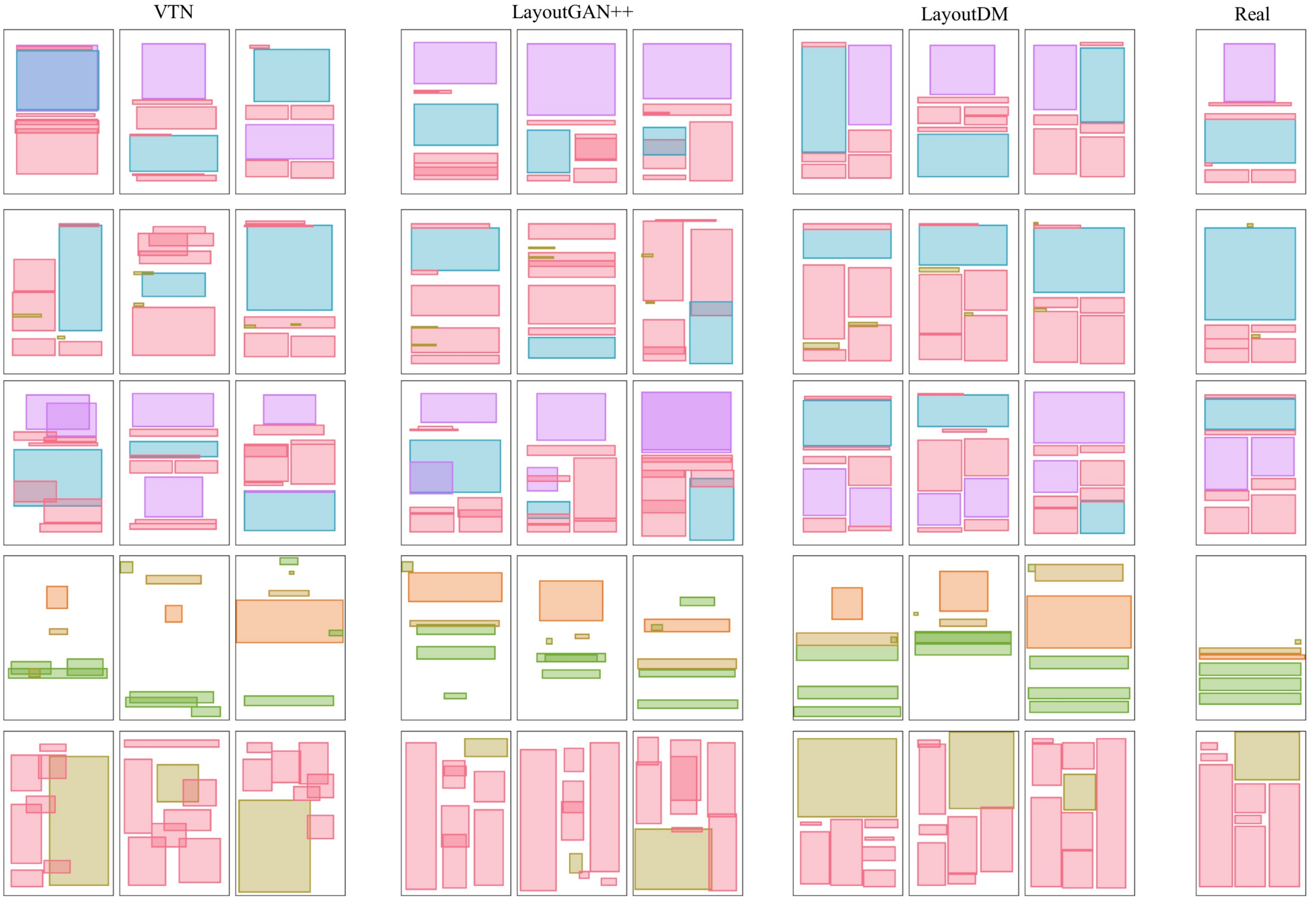

Figure 3. Additional conditional generation comparison. We show three samples for each model conditioned on the same element category labels. Real design are shown as Real.

References

[1] Diego Martín Arroyo, Janis Postels, and Federico Tombari. Variational transformer networks for layout generation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13637–13647, 2021. 2, 3

[2] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *2021 IEEE/CVF International Conference on*

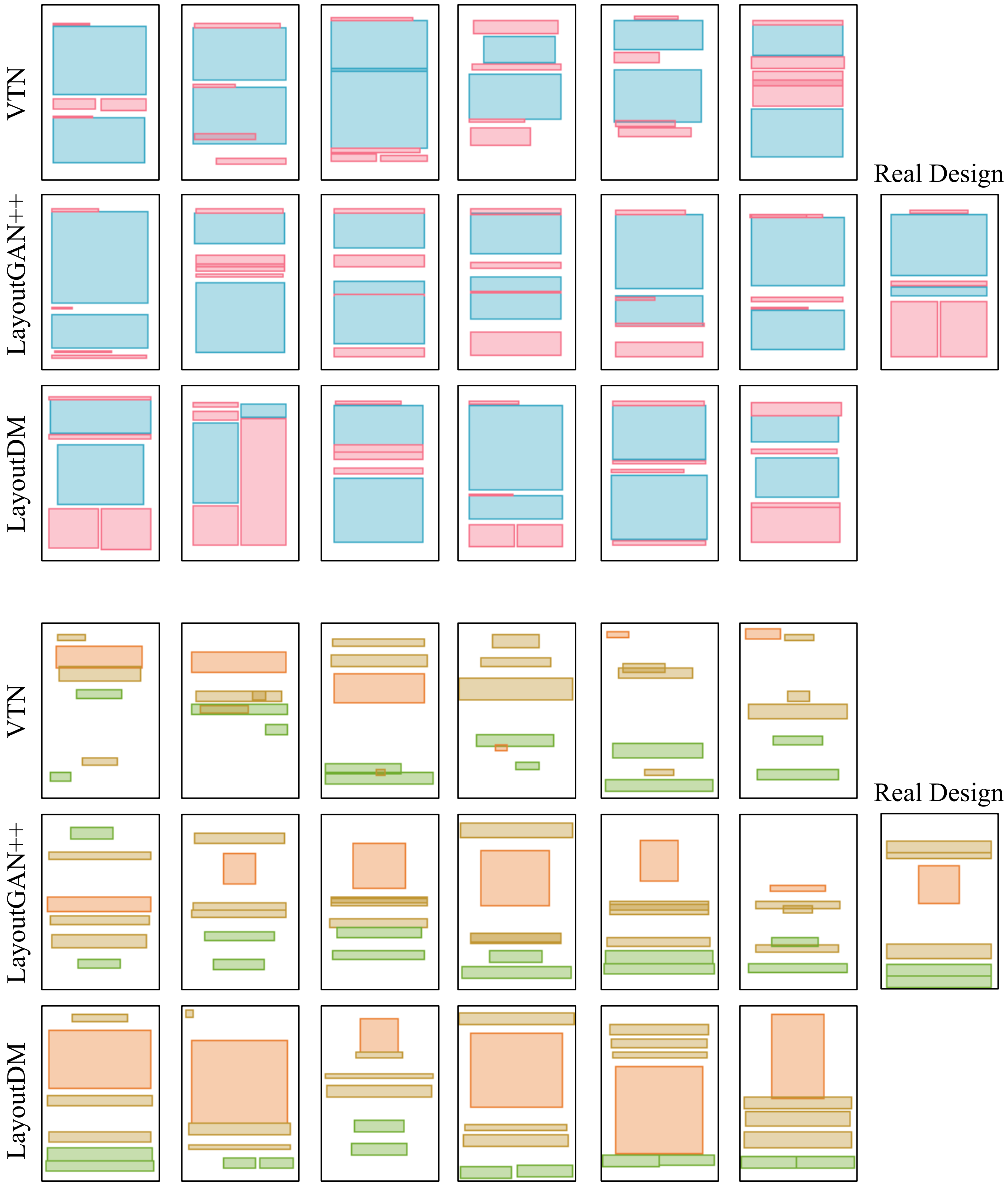

Figure 4. Additional quantitative diversity comparisons. The first to third rows are from the Publaynet dataset, and the fourth to sixth rows are from the Rico dataset. Layouts are generated conditioned on the same layout attributes (label categories) every three rows. Real designs are displayed for reference.

Computer Vision (ICCV), pages 984–994, 2021. 2, 3

[3] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9894–9903, 2019. 2, 3

[4] Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Constrained graphic layout generation via latent optimization. In *ACM International Conference on Multimedia*, MM '21, pages 88–96, 2021. 1, 2, 3

[5] Xiang Kong, Lu Jiang, Huiwen Chang, Han Zhang, Yuan Hao, Haifeng Gong, and Irfan Essa. Blt: bidirectional layout transformer for controllable layout generation. In *European Conference on Computer Vision*, pages 474–490. Springer, 2022. 1, 2, 3

[6] 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 Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 491–506, Cham, 2020. Springer International Publishing. 2, 3

[7] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. Layoutgan: Synthesizing graphic layouts with vector-wireframe adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2388–2399, 2021. 2

[8] Jianan Li, Jimei Yang, Jianming Zhang, Chang Liu, Christina Wang, and Tingfa Xu. Attribute-conditioned layout gan for automatic graphic design. *IEEE Transactions on Visualization and Computer Graphics*, 27(10):4039–4048, 2021. 1