BLT: Bidirectional Layout Transformer for Controllable Layout Generation

Xiang Kong[2⋆] , Lu Jiang[1], Huiwen Chang[1], Han Zhang[1],
Yuan Hao[1], Haifeng Gong[1], Irfan Essa[1,3]

[1] Google[2] LTI, Carnegie Mellon University, [3] Georgia Institute of Technology
xiangk@cs.cmu.edu, lujiang@google.com

1 Implementation Details

Training. To find out the optimal hyperparameters for each task, we use a grid search for the following ranges of possible values, learning rate in $\{1e-3, 3e-3, 5e-3\}$, dropout and attention dropout in $\{0.1, 0.3\}$. The data preprocessing procedure discussed in [1] is used. All baseline models, including ours, are trained on the same dataset by five independent trials, where the averaged metrics with standard deviations are reported.

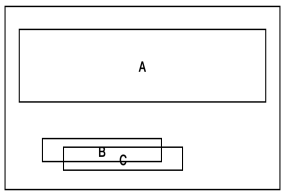


Fig. 1: A toy layout sample for the IOU computation. The metrics used yields more reasonable IOU $\frac{0.5}{6.5} = \frac{1}{13}$ than the IOU $\frac{0.5}{1.5} = \frac{1}{3}$ used in [1]

Notes on the evaluation metric IOU: In [1], the author calculates the IOU scores between all pairs of overlapped objects and average them. In our work, we propose to use the so-called perceptual IOU score which first projects the layouts as if they were images then computes the ovarlapped area divided by **the union area of all objects**. We show the difference via a toy example in Fig. 1. The areas of objects A, B and C are 5, 1, 1, the overlapped area of B and C are 0.5. Based on the IOU computation in [1], since they just care about overlapped objects, only the IOU of objects **B** and **C** are computed which is $\frac{0.5}{1.5} = \frac{1}{3}$. On the contrary, in our IOU computation, the overlapped area of B and C will be divided the union area of all objects, hence, the IOU of this layout is $\frac{0.5}{6.5} = \frac{1}{13}$ which is more reasonable than their result.

⋆ Work done during their research internship at Google.

2 Additional Quantitative Results

We show more results with more metrics on CoCo, Magazine and Ads datasets. Our proposed model consistently achieve better results on these datasets compared to autoregressive transformer-based models, demonstrating the effectiveness of our model.

COCO	Conditioned on Category				+ Size
Model	IOU↓	Overlap↓	Alignment↓	Sim.↑	Sim.↑
Trans.	0.60±0.4%	**1.66**±2.0%	0.34±0.2%	0.20±0.2%	-
VTN	0.63±0.4%	1.79±1.6%	0.32±0.3%	0.22±0.1%	-
Ours	**0.35**±0.5%	1.93±5.0%	**0.16**±0.5%	**0.24**±0.1%	**0.44**
Magazine	Conditioned on Category				+ Size
Model	IOU↓	Overlap↓	Alignment↓	Sim.↑	Sim.↑
Trans.	0.20±0.8%	0.22±1.6%	0.48±1.1%	0.15±0.3%	-
VTN	**0.18**±1.8%	0.15±1.2%	0.47±1.4%	0.15±0.9%	-
Ours	**0.18**±0.6%	**0.12**±1.8%	**0.44**±1.9%	**0.18**±0.4%	**0.27**
Ads	Conditioned on Category				+ Size
Model	IOU↓	Overlap↓	Alignment↓	Sim.↑	Sim.↑
Trans	0.19±0.1%	0.15±0.1%	0.35±0.1%	0.30±0.1%	-
VTN	0.18±0.2%	0.15±0.1%	0.33±0.1%	0.30±0.1%	-
Ours	**0.10**±0.4%	**0.10**±0.4%	**0.18**±0.6%	**0.31**±0.1%	0.41

Table 1: Category (+ Size) conditional layout generation performance on various benchmarks.

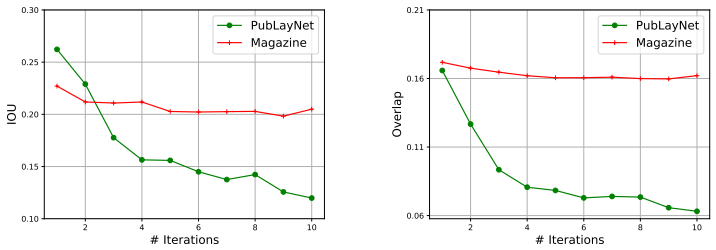


Fig. 2: IOU and overlap scores at different decoding iterations on two datasets.

3 Additional Visualization

3.1 Qualitative Results for Conditional Generation

We show samples in Fig. 3 from conditional generation on category and size for four design applications including the mobile UI interface, scientific paper, magazine and natural scenes. We also show some samples in comparison with Layout-VAE [2] and NDN [3] in Fig. 4.

3.2 Examples of Diverse Conditional Generation

In our main experiment, we use greedy search to find out the most likely candidate for each attribute at each iteration. Here, we generate layouts through sampling the top-k ($k = 10$) from the likelihood distribution for category + size conditional generation. This leads to diverse layouts. Some examples are shown in Fig. 6.

3.3 More Attention Head Patterns

Patterns for other heads at different layers are listed in Fig. 7. We could find that for masked x position (head 1-1 and head 2-6, *etc.*), their heads will attend to width information of various objects for accurate prediction. And similar findings could be found for other heads.

3.4 Failure Cases

Some undesired conditional generation results are shown in Fig 5. Similar to other layout generation models, there are some overlaps between objects in some generation results. Furthermore, some generated samples are largely different from the real layouts with low visual quality. For example, in the second sample on the Magazine, the alignment of the generated sample is worse than its corresponding real layout. We will explore these directions in the future work.

3.5 Iterative Refinement Process

To understand the process of our iterative refinement algorithm, we explore the performance of models with various iterations. Quantitatively, IOU and Overlap metrics, where the lower, the better, are plotted in Fig.2 along with the number of iterations for refinement. With more iterations, the quality metrics are getting improved and stable. We also show samples of generated layouts at different number of iterations in Fig. 8. At the first iteration, there are severe overlaps between objects, showing the difficulty to yield high-quality layouts with just one pass. However, after iteratively refining low-confident attributes, the layouts become more realistic.

Fig. 3: Conditional layout generation for scientific papers, user interface, and magazine. The user inputs are the object category and their size (width, height). We compare the generated layout and the real layout with the same input in the dataset.

Fig. 4: Qualitative Results for conditional generation on PublayNet and RICO from BLT and VAE-based models.

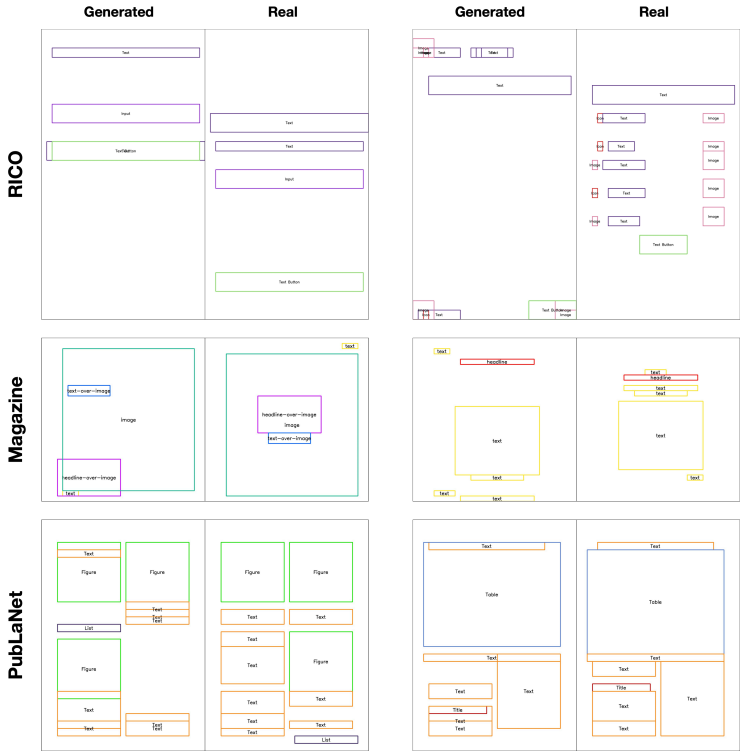


Fig. 5: Failure cases for layout generation using the propose method. We compare the generated layout and the real layout with the same input in the dataset. See Section 3.4 for more discussion.

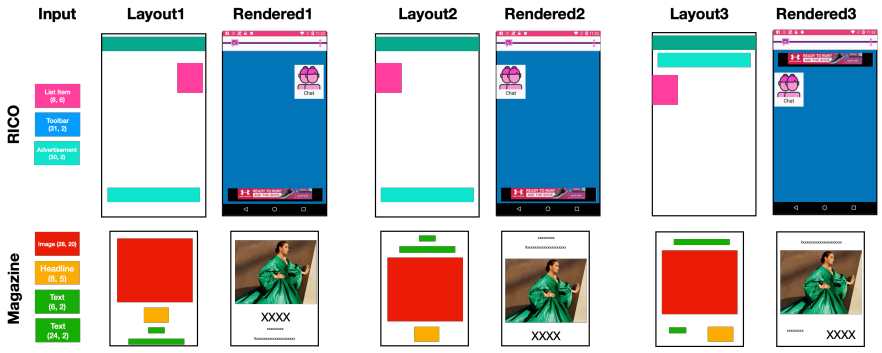
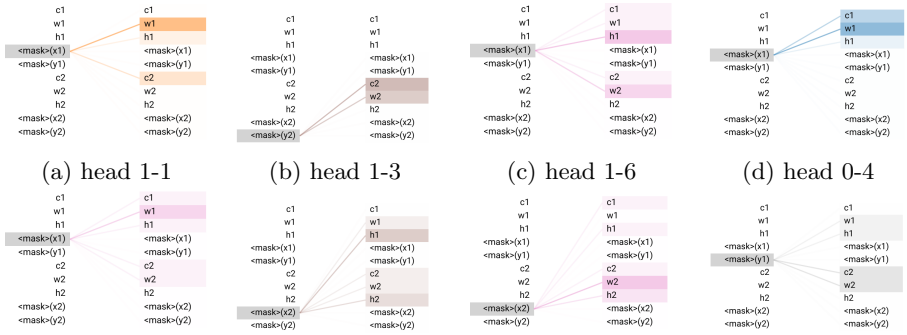


Fig. 6: Diverse conditional generation via top-k sampling method.



(a) head 1-1 (b) head 1-3 (c) head 1-6 (d) head 0-4

(e) head 2-6 (f) head 2-5 (g) head 3-6 (h) head 3-7

Fig. 7: Additional examples of attention heads exhibiting the patterns for masked tokens. The darkness of a line indicates the strength of the attention weight (some attention weights are so low they are invisible). We use ⟨layer⟩-⟨head number⟩ to denote a particular attention head.

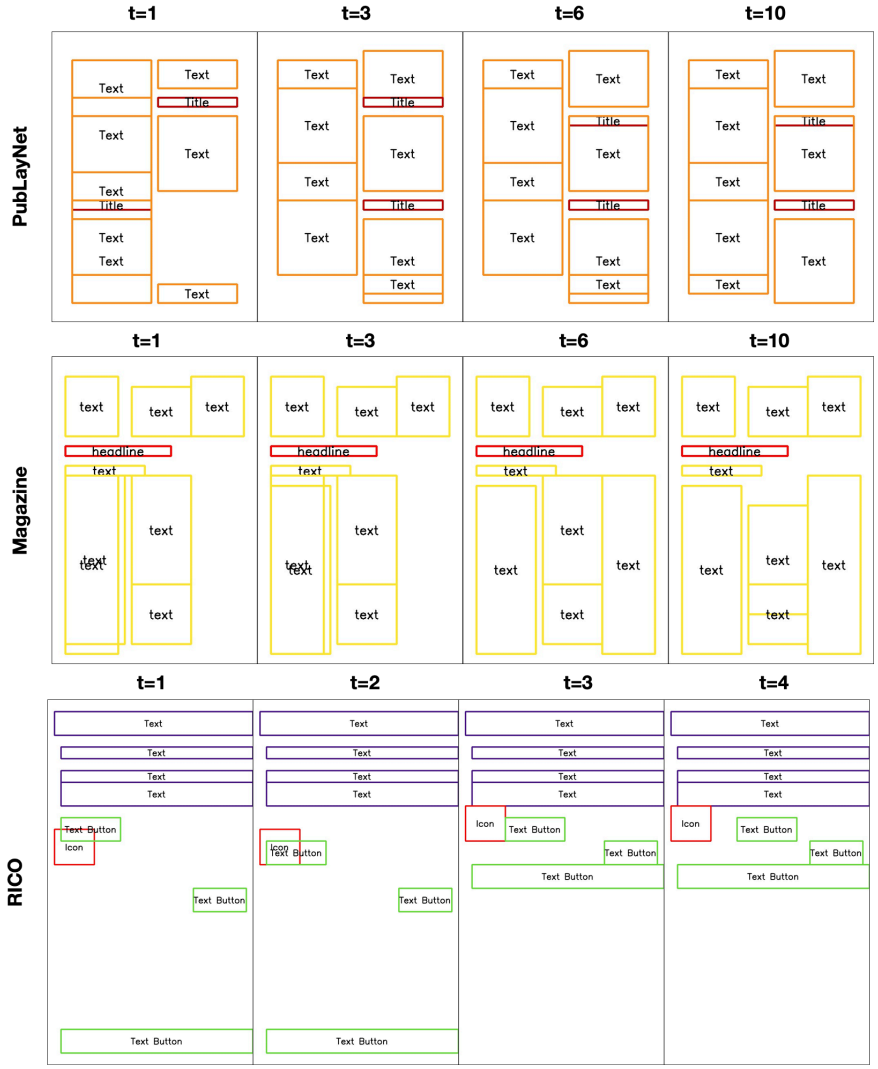


Fig. 8: Layouts refinement process. Layouts generated at different iterations (t) are shown on three datasets.

References

1. Arroyo, D.M., Postels, J., Tombari, F.: Variational transformer networks for layout generation. In: CVPR (2021)
2. Jyothi, A.A., Durand, T., He, J., Sigal, L., Mori, G.: Layoutvae: Stochastic scene layout generation from a label set. In: CVPR (2019)
3. Lee, H.Y., Jiang, L., Essa, I., Le, P.B., Gong, H., Yang, M.H., Yang, W.: Neural design network: Graphic layout generation with constraints. In: ECCV (2020)