Supplementary Material for "LayoutFormer++: Conditional Graphic Layout Generation via Constraint Serialization and Decoding Space Restriction"

1. Implementation Details

1.1. Training

The training epochs and batch sizes of LayoutFormer++ for each layout generation task are shown in Table 1. On both RICO and PubLayNet, we use optimizer Adafactor with a learning rate of 0.0001. We use the learning rate warmup. The numbers of the warmup steps for each model are also shown in Table 1.

	RICO			PubLayNet		
Task	Epoch	Batch Size	Warmup Steps	Epoch	Batch Size	Warmup Steps
Gen-T	100	32	1000	60	100	1000
Gen-TS	100	32	1000	60	100	4000
Gen-R	150	16	1000	60	64	3000
Completion	100	32	1000	60	100	3000
Refinement	100	32	1000	60	100	2000
UGen	100	32	1000	60	100	1000

Table 1. Hyper-parameters for training LayoutFormer++.

1.2. Inference

During the inference stage, we leverage top-k sampling with $k = 10$ and temperature $\tau = 0.7$ for LayoutFormer++ to generate diverse layouts for all tasks, except refinement which simply uses greedy sampling. For the decoding space restriction strategy, to avoid the decoder make the same prediction after backtracking, we change the top-k temperature τ from 0.7 to 1.5 to smooth the sampling distribution at the step that the decoding process backs to. The max back time $maxBack$ of the backtracking mechanism, and the threshold θ of the probability pruning module, are tuned to achieve the best performance for each task (i.e., Gen-T, Gen-TS, and Gen-R) and dataset, which are shown in Table 2.

	RICO			PubLayNet		
Tasks	Gen-T	Gen-TS	Gen-R	Gen-T	Gen-TS	Gen-R
$maxBack$	5	5	5	5	3	3
θ	0.8	0.3	0.2	0.3	0.3	0.2

Table 2. Hyper-parameters of the Decoding Space Restriction Strategy during the Inference

2. More Discussion on Constraint Serialization

2.1. All-in-one Layout Generation Model

The unification of the constraint format and the model architecture opens up a chance for training a single model to serve all tasks simultaneously. This is of great benefit. It saves a lot of deployment efforts in practice since there is a single set of model weights for flexibly serving multiple tasks.

To achieve this goal, we train LayoutFormer++ for the mixed-task generation, denoted as LayoutFormer++(M). Following [1–3], we prepend a task indicator token to each input sequence and combine the training data of all tasks with temperature mixing. The data sample weights of each task for constructing a mixed dataset are shown in Table 3, which are the same on both Rico and PubLayNet. In practice, we find that slightly increasing the weights of Refinement and Gen-TS can better balance the performance for all tasks. The task loss weights are all set as 1.

Tasks	Gen-T	Gen-TS	Gen-R	Refinement	Completion	UGen
Weights	$\frac{1}{12}$	$\frac{1}{3}$	$\frac{1}{12}$	$\frac{1}{3}$	$\frac{1}{12}$	$\frac{1}{12}$

Table 3. Task sample weights in constructing mixed dataset for LayoutFormer++(M).

The quantitative comparison between LayoutFormer++(M) with the state-of-the-art baselines of each task are shown in Table 4. We highlight the results of LayoutFormer++(M) by bold when they achieve better performance than the baselines. We find that although handling all the tasks simultaneously is much more difficult than tackling one task, our LayoutFormer++(M) still significantly outperforms the baselines on most metrics. This indicates that our LayoutFormer++ can be trained as an all-in-one model, to flexibly handle different layout generation tasks by a single set of model parameters.

		RICO				PubLayNet			
Tasks	Methods	mIoU ↑	FID ↓	Align. ↓	Overlap ↓	mIoU ↑	FID ↓	Align. ↓	Overlap ↓
Gen-T	LayoutGAN++	0.298	5.954	0.261	0.620	0.297	14.875	0.124	0.148
	LayoutFormer++(M)	**0.396**	**2.101**	**0.161**	**0.586**	**0.352**	**10.620**	**0.021**	**0.018**
Gen-TS	BLT	0.604	0.951	0.181	0.660	0.428	7.914	0.021	0.419
	LayoutFormer++(M)	0.577	1.392	**0.179**	**0.567**	**0.463**	**2.097**	0.026	**0.041**
Gen-R	CLG-LO	0.286	8.898	0.311	0.615	0.277	19.738	0.123	0.200
	LayoutFormer++(M)	**0.372**	11.026	**0.122**	**0.593**	**0.318**	**11.694**	**0.024**	**0.123**
Refinement	RUITE	0.811	0.107	0.133	0.483	0.781	0.061	0.029	0.020
	LayoutFormer++(M)	0.786	**0.084**	0.135	0.495	0.773	0.094	**0.022**	**0.006**
Completion	LayoutTransformer	0.363	6.679	0.194	0.478	0.077	14.769	0.019	0.0013
	LayoutFormer++(M)	**0.731**	**4.104**	**0.074**	**0.472**	**0.475**	**8.304**	0.023	0.0016
UGen	LayoutTransformer	0.439	22.884	0.052	0.471	0.062	36.304	0.031	0.0009
	LayoutFormer++(M)	**0.734**	**11.667**	0.058	**0.463**	**0.430**	**30.161**	**0.029**	**0.0008**

Table 4. Quantitative comparisons between LayoutFormer++(M) and task-specific baselines on six tasks and two datasets.

2.2. Generalization to New Tasks

In this section, we aim to demonstrate that LayoutFormer++ can be flexibly adapted to the new layout generation tasks. We propose four new tasks to develop the experiments: *Gen-TC*, *Gen-TSC*, *Gen-RS* and *Gen-RP*, by combining the existing typical tasks:

Gen-TC is the combination of Gen-T and Completion. In this task, user has already placed some elements on the layout, while specifying the types of other elements that are required to be arranged. We formulate the input as $S_{\text{Gen-TC}} = \{\langle \text{sos} \rangle c_1 x_1 y_1 w_1 h_1 | \dots | c_P x_P y_P w_P h_P | | c_{P+1} | \dots | c_N \langle \text{eos} \rangle\}$, where P is the number of already placed elements, and N denotes the total number of elements.

Gen-TSC is the combination of Gen-TS and Completion. In this tasks, user has already placed some elements on the layout, while specifying the types and sizes of other elements that are required to be arranged. We formulate the input as $S_{\text{Gen-TSC}} = \{\langle \text{sos} \rangle c_1 x_1 y_1 w_1 h_1 | \dots | c_P x_P y_P w_P h_P || c_{P+1} w_{P+1} h_{P+1} | \dots | c_N w_N h_N \langle \text{eos} \rangle\}$.

Gen-RS combines the Gen-R with Gen-TS. In this task, user specifies the types and sizes of the elements, and also require the relative position relationships (i.e., above, bottom, left, right, and overlap) between some elements. The input constraint sequence is formulated as $S_{\text{Gen-RS}} = \{\langle \text{sos} \rangle c_1 w_1 h_1 | c_2 w_2 h_2 | \dots | c_N w_N h_N || c_{k_1} k_1 r_{k_1,k_2} c_{k_2} k_2 | \dots | c_{k_{2M-1}} k_{2M-1} r_{k_{2M-1},k_{2M}} c_{k_{2M}} k_{2M} \langle \text{eos} \rangle\}$, where M is the number of the positional relationships.

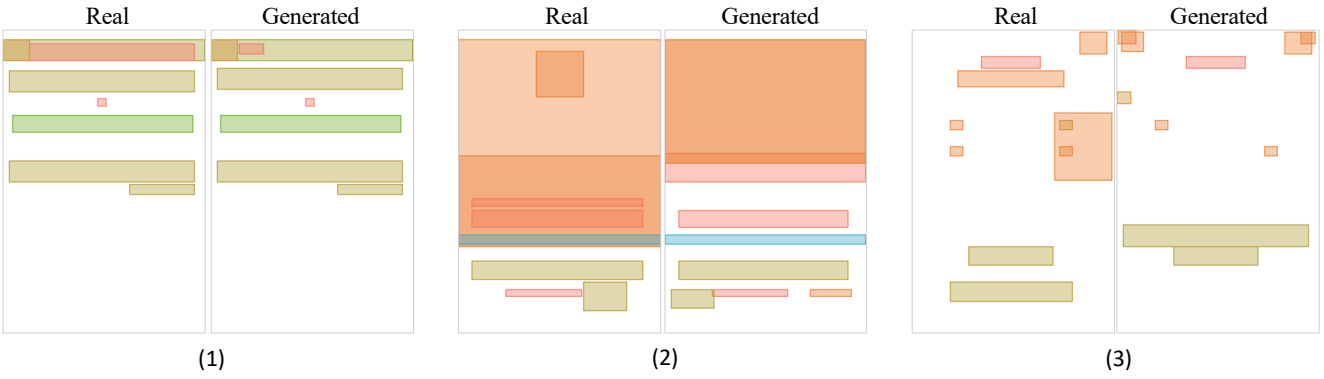
Gen-RP generates layouts from user-specified element types and positions, and relative size relationships (i.e., smaller, larger and equal) between the elements. The input constraint sequence is formulated as $S_{\text{Gen-RP}} = \{\langle \text{sos} \rangle c_1 x_1 y_1 | c_2 x_2 y_2 | \dots | c_N x_N y_N || c_{k_1} k_1 r_{k_1,k_2} c_{k_2} k_2 | \dots | c_{k_{2M-1}} k_{2M-1} r_{k_{2M-1},k_{2M}} c_{k_{2M}} k_{2M} \langle \text{eos} \rangle\}$, where M is the number of the size relationships.

We leverage LayoutFormer++ for the four new tasks on the Rico dataset. For Gen-TC and Gen-TSC, we random sample 50% elements of each layout as the given complete elements. For Gen-RS and Gen-RP, same as in Gen-R, we randomly sample 10% element relationships as the input.

Table 5 shows the quantitative results of these combined new tasks. Since there is no baseline for the new tasks, we just give the results of the related typical tasks at the bottom of Table 5 for reference. Figure 1 shows the qualitative results for the combined tasks. For each task, we show three groups of layouts, where the right one in each group is the generation by LayoutFormer++, and the left one is the real layout where the input constraints for inference come from. It can be obtained that LayoutFormer++ can flexibly handle all these new layout generation tasks with achieving good generation quality.

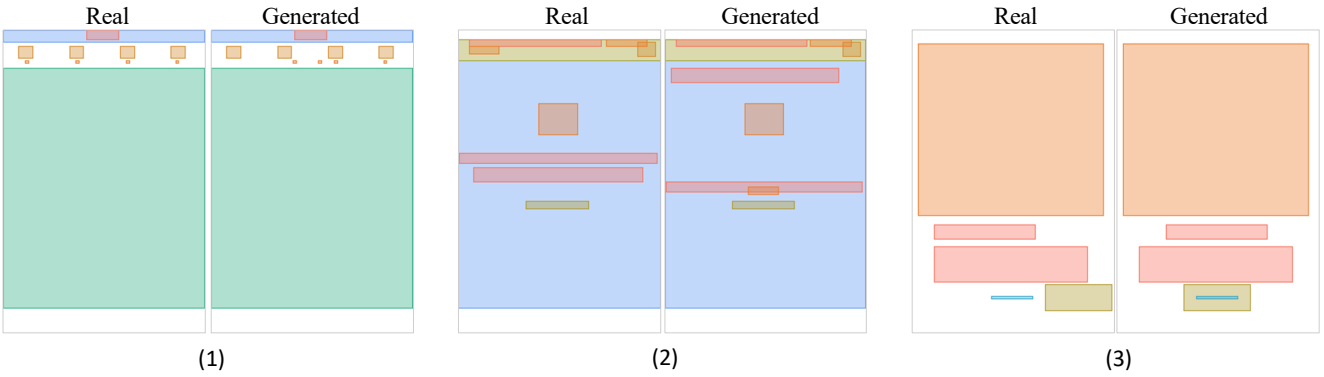
	Tasks	mIoU ↑	FID ↓	Align. ↓	Overlap ↓
Combined Tasks	Gen-TC	0.744	0.862	0.093	0.562
	Gen-TSC	0.818	0.534	0.129	0.529
	Gen-RS	0.593	5.022	0.190	0.561
	Gen-RP	0.799	0.769	0.119	0.550
Typical Tasks	Gen-T	0.432	1.096	0.230	0.530
	Gen-TS	0.620	0.757	0.202	0.542
	Gen-R	0.424	5.972	0.332	0.537

Table 5. Quantitative evaluation for the combined tasks on Rico.

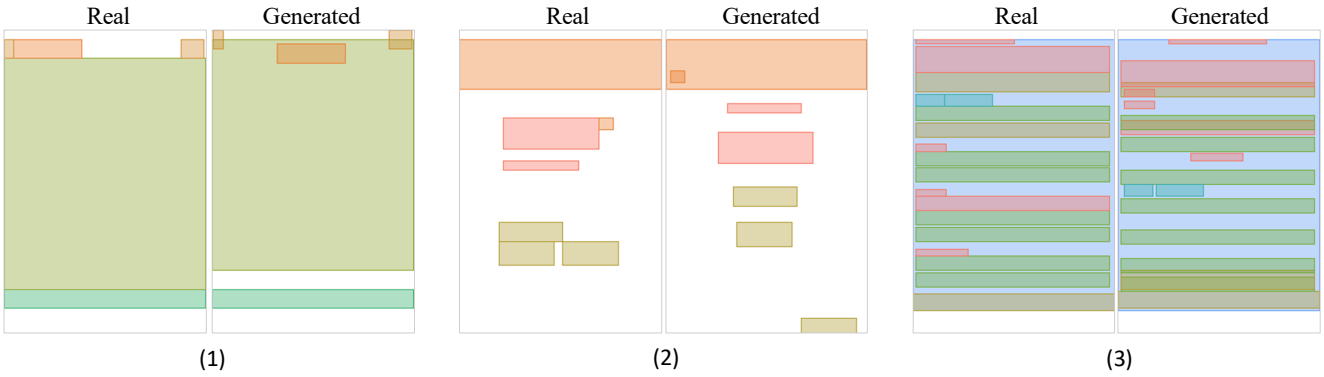


(a) Gen-TC

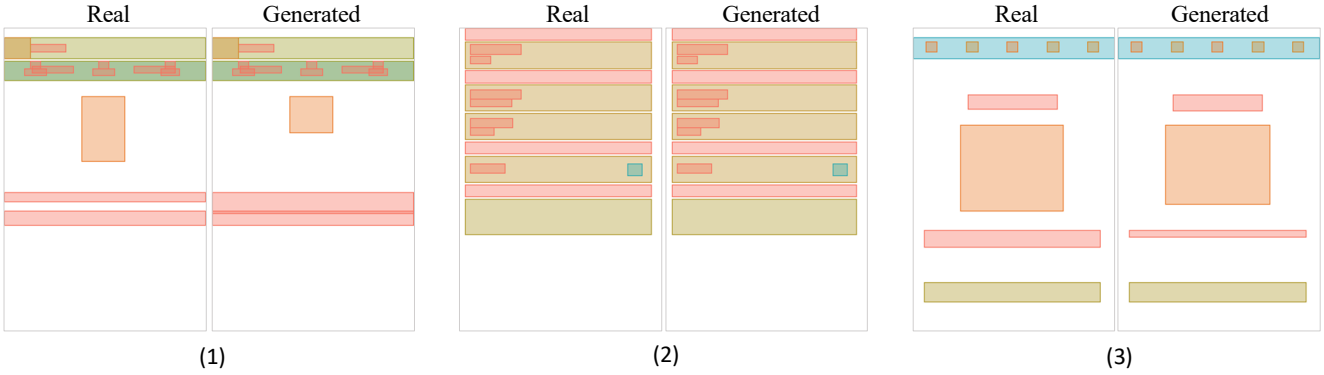
Figure 1. Qualitative results of Gen-TC, Gen-TSC, Gen-RS and Gen-RP.



(b) Gen-TSC



(c) Gen-RS



(d) Gen-RP

Figure 1. (Cont.) Qualitative results of Gen-TC, Gen-TSC, Gen-RS and Gen-RP.

3. More Discussion on Decoding Space Restriction

As we introduced in the paper, the decoding space restriction strategy prunes the predicted distribution at each decoding step, and restarts the decoding process from a previous step when current prediction cannot achieve good controllability. Another naive and straightforward approach is running the decoding process multiple times and then selecting the best generation among them as the final results. In this section, we compare the decoding space restriction strategy with this naive approach.

We develop the comparison on the Gen-R tasks. For the naive approach, we run the decoding process for 5 and 3 times on

Rico and PubLayNet datasets respectively, which is same as the number of the max back times we set for the backtracking mechanism. We try different metrics to select the result from multi-times decoding for evaluation. *select by Align.* denotes that for each layout, the result with the minimum value of align. is chosen for developing the evaluation. *select by Overlap* denotes the result is selected by the overlap value, and *select by Vio.%* denotes the results is selected by the Vio.%.

Table 6 shows the quantitative comparison. We have following observations. For *select by Align.* and *select by Overlap*, they significantly improve the performance on Align. and Overlap respectively. However, they do not perform well on Vio %. *select by Vio.%* achieves better Vio % than *select by Align.* and *select by Overlap*. However, LayoutFormer++ significantly outperforms *select by Vio.%* on Vio.%, while achieving comparable performance on the quality metrics with all the baselines. This demonstrates the advantage of the decoding space restriction strategy. Without the restriction to the predicted distribution, the naive approach is very difficult to sample the attributes for the layout to conform all the constraints, even sample multiple times.

		RICO					PubLayNet				
Tasks	Methods	mIoU ↑	FID ↓	Align. ↓	Overlap ↓	Vio. % ↓	mIoU ↑	FID ↓	Align. ↓	Overlap ↓	Vio. % ↓
Gen-R	select by Align.	0.454	5.708	0.096	0.560	32.97	0.359	4.640	0.0007	0.029	16.05
	select by Overlap	0.454	6.466	0.336	0.422	32.66	0.352	5.397	0.023	0.0003	15.74
	select by Vio.%	0.471	4.552	0.316	0.542	27.45	0.361	4.344	0.022	0.027	6.6
	LayoutFormer++	0.424	5.972	0.332	0.537	**11.84**	0.353	4.954	0.025	0.076	**3.9**

Table 6. Quantitative comparisons with the naive approach.

4. More Qualitative Results

We show more generated layouts here to better demonstrate the good performance of LayoutFormer++. For all the six typical layout generation tasks, we present the generated layouts by LayoutFormer++ on both RICO and PubLayNet.

Gen-T. The generated layouts of Gen-T are shown in Figure 2. We show six groups of generated results on both Rico and PubLayNet, each group contains two layouts generated from the same element type constraints below.

Gen-TS. The generated layouts of Gen-TS on Rico and PubLayNet are shown in Figure 3. Similar with the results of Gen-T, each group contains two layouts generated from the same element type and size constraints in the below table.

Gen-R. The generated layouts of Gen-R on Rico and PubLayNet are shown in Figure 4. We list the relationship constraints that the layouts generated from in the below table.

Refinement. The generated layouts of Refinement on Rico and PubLayNet are shown in Figure 5. We show six groups of generated layouts, where each group contains two layouts. The left one is the noised layout that needs refinement, and the right one is the layout refined by LayoutFormer++.

Completion. The generated layouts of Completion are shown in Figure 6. We show four groups of generated layouts on both Rico and PubLayNet, where each group contains three layouts generated from the same first element. By leveraging top-k sampling, LayoutFormer++ can complete the same partial layout into various final layouts.

UGen. The generated layouts of UGen on RICO and PubLayNet are shown in Figure 7. The top-k sampling ensures the diversity of generated layouts from identical input for LayoutFormer++.

References

[1] Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*, 2022. 2

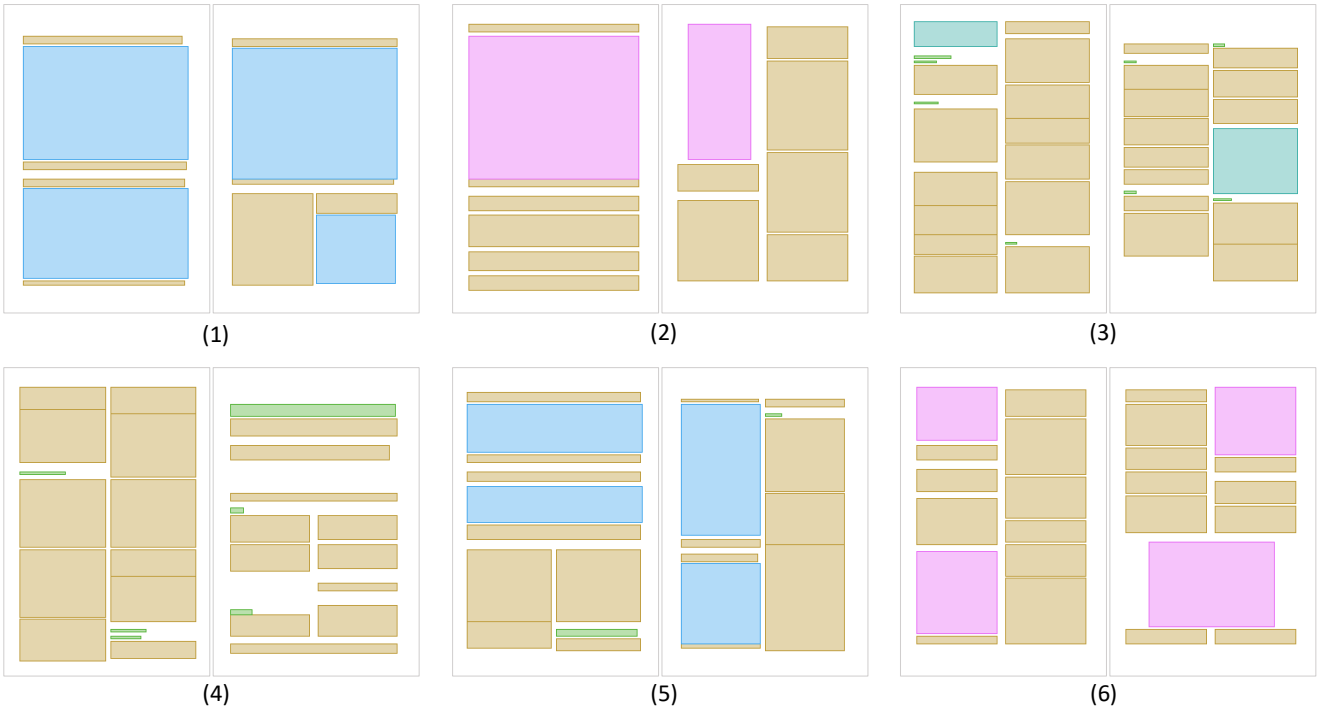
[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 2

[3] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. 2

	Element Type Constraints
(1)	Text, Image, Text Button*2
(2)	Text*5, Image*6, Icon*4, Text Button*3
(3)	Text*2, Radio Button*2, Icon*2, Text Button, Toolbar, Card
(4)	Text*2, Icon, Text Button*2, Toolbar, Input
(5)	Text, Image, Text Button*6, Input*2
(6)	Text*10, Image, Icon, List Item*5, Text Button, Toolbar

(a) Rico

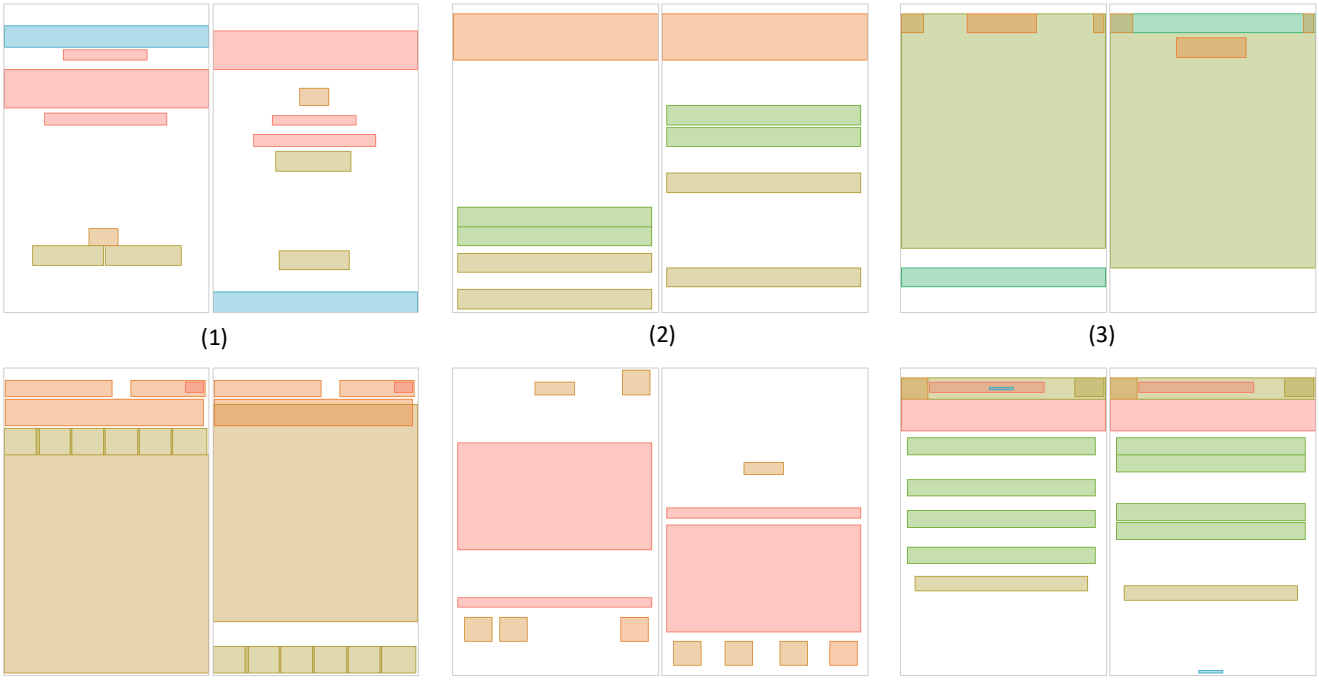
Figure 2. Qualitative results of Gen-T on Rico and PubLayNet. The element type constraints are in the table.



Element Type Constraints	
(1) text*4, table*2	(4) text*11, title*3
(2) text*6, figure	(5) text*8, title, table*2
(3) text*13, title*4, list	(6) text*10, figure*2

(b) PubLayNet

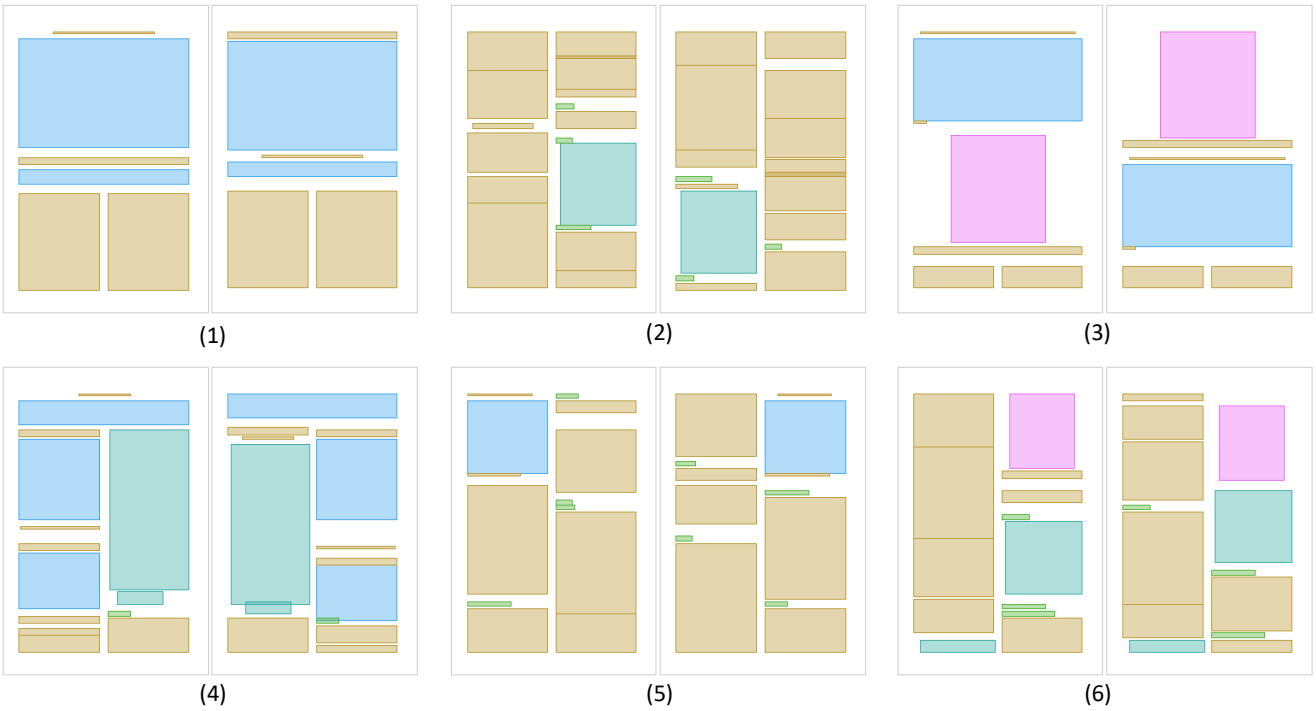
Figure 2. (Cont.) Qualitative results of Gen-T on Rico and PubLayNet. The element type constraints are in the table.



Element Type and Size Constraints	
(1)	Text 127 16, Text 52 4, Text 76 5, Pager Indicator 127 9, Icon 18 7, Text Button 47 8, Text Button 44 8
(2)	Image 127 19, Text Button 120 8, Text Button 120 8, Input 120 8, Input 120 8
(3)	Advertisement 127 8, Image 43 8, Icon 6 8, Icon 14 8, Web View 127 97
(4)	Text 11 4, Image 123 11, Image 66 7, Image 46 7, List Item 127 90, Text Button 22 11, Text Button 22 11, Text Button 22 11, Text Button 22 11, Text Button 22 11, Text Button 22 11
(5)	Text 120 4, Text 120 44, Image 17 10, Icon 17 10, Icon 24 5, Icon 17 10, Icon 17 10
(6)	Text 127 13, Text 71 4, Pager Indicator 15 1, Icon 17 9, Text Button 18 8, Text Button 107 6, Toolbar 127 9, Input 117 7, Input 117 7, Input 117 7, Input 117 7

(a) Rico

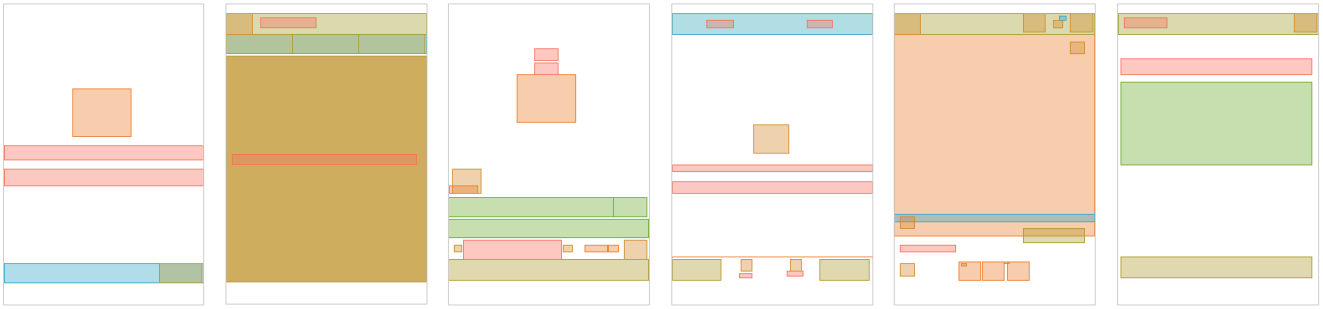
Figure 3. Qualitative results of Gen-TS on Rico and PubLayNet. The tables show the element type and size constraints.



Element Type and Size Constraints	
(1)	text 63 1, text 105 3, text 50 40, text 50 40, table 105 6, table 105 45
(2)	text 50 16, text 50 20, text 50 35, text 50 11, text 50 14, text 50 11, text 38 2, text 50 7, text 50 3, text 50 16, text 50 16, text 50 7, title 11 2, title 22 2, title 10 2, list 47 34
(3)	text 50 9, text 50 9, text 97 1, text 8 1, text 105 3, table 105 34, figure 59 44
(4)	text 49 1, text 50 3, text 32 1, text 50 3, text 50 3, text 50 14, text 50 7, text 50 3, title 14 2, list 49 66, list 28 5, table 50 33, table 105 10, table 50 23
(5)	text 33 1, text 40 1, text 50 18, text 50 16, text 50 42, text 50 45, text 50 5, text 50 26, title 27 2, title 14 2, title 10 2, title 12 2, table 50 30
(6)	text 50 38, text 50 24, text 50 14, text 50 3, text 50 14, text 50 5, text 50 22, title 33 2, title 17 2, title 27 2, list 47 5, list 48 30, figure 40 31

(b) PubLayNet

Figure 3. (Cont.) Qualitative results of Gen-TS on Rico and PubLayNet. The tables show the element type and size constraints.

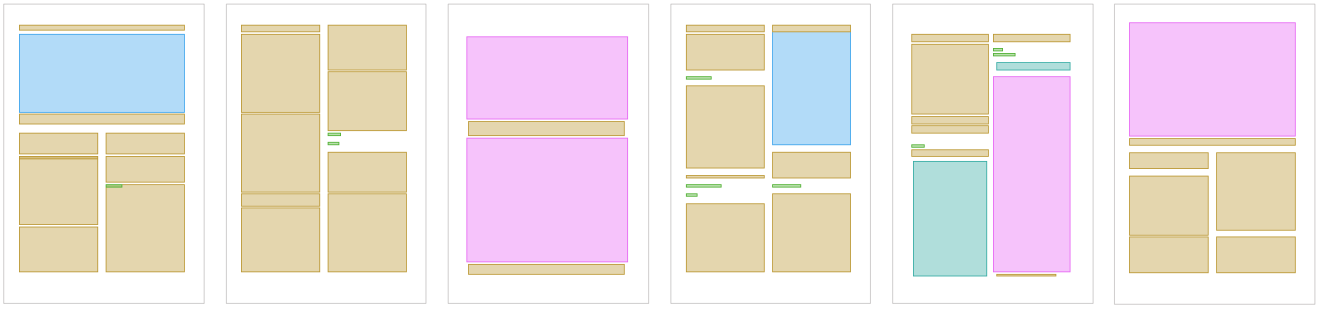


(1) (2) (3) (4) (5) (6)

Element Relationship Constraints	
(1)	Text*2, Pager Indicator, Image, Text Button ,Image 1 top Text 2, Text Button 1 bottom Image 1
(2)	Text*2, Multi-Tab, Icon, List Item*3, Text Button*3, Toolbar ,Text Button 1 top canvas, List Item 1 larger Text 2, List Item 2 center Text 2, Text Button 3 top Text 2, Icon 1 smaller Multi-Tab 1, List Item 2 larger Multi-Tab 1, List Item 3 center List Item 1, Text Button 1 top List Item 3, Text Button 2 top List Item 3, Text Button 3 smaller List Item 3, Text Button 3 right Text Button 1, Toolbar 1 top Text Button 1
(3)	Text*3, Image*4, Icon*4, Text Button, Input*3, Text Button 1 larger Text 1, Icon 2 larger Text 2, Icon 3 smaller Text 2, Icon 4 smaller Text 2, Icon 4 bottom Text 2, Input 1 bottom Text 2, Image 3 smaller Text 3, Icon 3 smaller Text 3, Icon 4 smaller Text 3, Input 2 smaller Text 3, Icon 3 smaller Image 1, Text Button 1 larger Image 1, Icon 1 larger Image 2, Icon 3 bottom Image 2, Icon 2 larger Image 3, Icon 2 right Image 3, Icon 3 smaller Image 3, Icon 2 larger Image 4, Icon 2 smaller Icon 1, Input 3 larger Icon 1, Input 3 bottom Icon 1, Icon 4 equal Icon 3, Input 1 top Icon 3
(4)	Text*6, Multi-Tab, Image, Icon*3, Text Button*2, Image 1 bottom canvas, Text Button 2 bottom canvas, Text 5 smaller Text 1, Text 6 bottom Text 1, Multi-Tab 1 center Text 1, Text 5 smaller Text 2, Icon 1 larger Text 2, Icon 3 smaller Text 2, Text Button 2 larger Text 2, Icon 3 bottom Text 3, Text 6 smaller Text 4, Text Button 2 right Text 6, Text Button 2 bottom Multi-Tab 1, Icon 1 larger Image 1, Icon 2 bottom Image 1, Icon 3 equal Icon 2, Text Button 1 larger Icon 2, Text Button 1 left Icon 2
(5)	Text, Pager Indicator*2, Image*4, Icon*9, Text Button, Toolbar, Text 1 bottom canvas, Icon 7 bottom canvas, Toolbar 1 top canvas, Text Button 1 larger Text 1, Icon 2 larger Pager Indicator 1, Toolbar 1 center Pager Indicator 1, Image 2 bottom Pager Indicator 2, Image 3 bottom Pager Indicator 2, Icon 6 center Pager Indicator 2, Image 3 bottom Image 1, Icon 3 smaller Image 1, Icon 6 center Image 1, Icon 8 smaller Image 1, Icon 9 smaller Image 1, Icon 3 equal Image 2, Icon 4 top Image 2, Icon 8 equal Image 2, Icon 2 top Image 3, Icon 1 larger Image 4, Icon 3 larger Image 4, Icon 7 left Image 4, Icon 8 larger Image 4, Icon 3 right Icon 1, Icon 6 smaller Icon 1, Icon 7 bottom Icon 1, Text Button 1 larger Icon 2, Toolbar 1 center Icon 2, Icon 6 bottom Icon 4, Icon 9 larger Icon 4, Icon 8 equal Icon 7, Icon 9 left Icon 8, Toolbar 1 top Text Button 1
(6)	Text*2, Icon, Text Button, Toolbar, Input, Text 2 bottom Text 1, Text Button 1 bottom Text 2, Text Button 1 larger Icon 1

(a) Rico

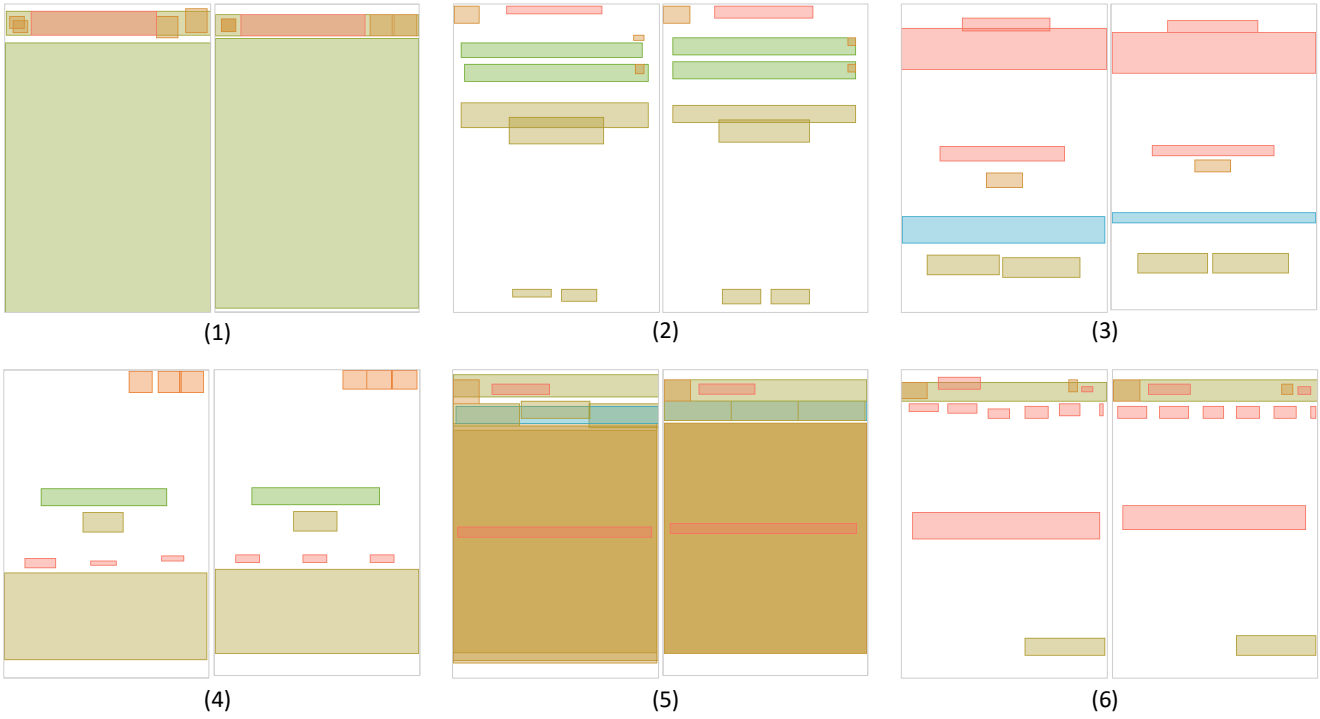
Figure 4. Qualitative results of Gen-R on Rico and PubLayNet. The tables show the element relationship constraints.



Element Relationship Constraints
(1) text*9, title, table, text 2 bottom text 1, title 1 bottom text 1, text 3 smaller text 2, text 4 smaller text 2, text 4 top text 2, text 8 smaller text 2, text 8 top text 2, title 1 left text 2, text 6 top text 5, text 9 bottom text 5, table 1 top text 6
(2) text*9, title*2, text 6 right text 1, text 7 larger text 1, text 8 larger text 1, text 9 bottom text 1, text 6 top text 2, text 7 larger text 2, text 8 left text 3, text 7 bottom text 4, text 9 bottom text 4, text 9 left text 5, text 7 equal text 6, text 8 top text 7
(3) text*2, figure*2, figure 1 top canvas, figure 1 top text 2
(4) text*8, title*4, table, text 6 center canvas, text 3 larger text 1, text 7 top text 1, title 4 smaller text 1, text 3 larger text 2, table 1 bottom text 2, text 6 larger text 3, title 3 smaller text 4, title 4 smaller text 4, text 7 smaller text 5, title 1 top text 5, title 2 smaller text 5, title 3 smaller text 5, table 1 smaller text 5, title 1 top text 6, title 4 smaller title 1, title 4 larger title 3, title 4 top title 3
(5) text*7, title*3, list*2, figure, text 4 center text 1, text 5 smaller text 2, title 2 smaller text 2, list 2 larger text 2, figure 1 top text 2, list 2 left text 3, list 2 larger text 4, title 2 smaller text 5, figure 1 larger text 5, title 1 smaller text 6, title 1 bottom text 7, list 1 smaller text 7, list 1 top text 7, list 1 top title 1, figure 1 larger title 3
(6) text*6, figure, text 2 top text 1, text 3 left text 2, text 5 larger text 4

(b) PubLayNet

Figure 4. (Cont.) Qualitative results of Gen-R on Rico and PubLayNet. The tables show the element relationship constraints.



(a) Rico

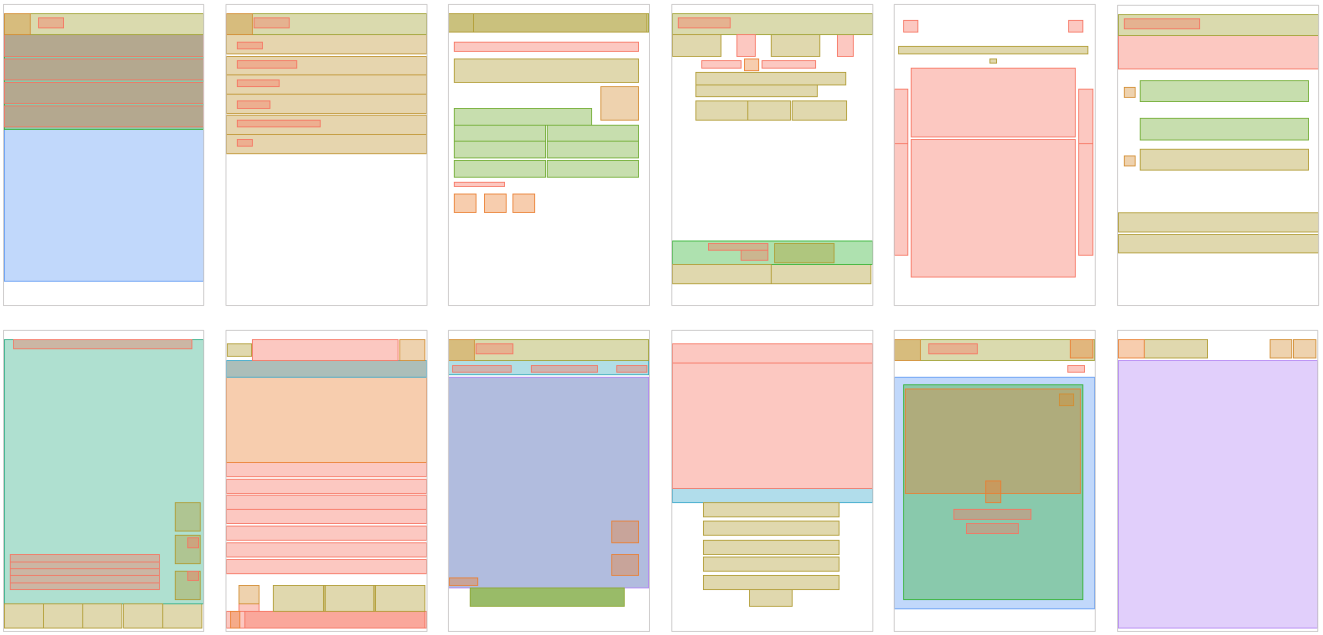
(b) PubLayNet

Figure 5. Qualitative results of Refinement on Rico and PubLayNet.

(a) Rico

(b) PubLayNet

Figure 6. Qualitative results of Completion on Rico and PubLayNet.



(a) Rico

(b) PubLayNet

Figure 7. Qualitative results of UGen on Rico and PubLayNet.