

# SciPostLayout: A Dataset for Layout Analysis and Layout Generation of Scientific Posters

Shohei Tanaka<sup>1</sup>  
shohei.tanaka@sinicx.com

Hao Wang<sup>2,1</sup>  
conan1024hao@akane.waseda.jp

Yoshitaka Ushiku<sup>1</sup>  
yoshitaka.ushiku@sinicx.com

<sup>1</sup> OMRON SINIC X Corporation  
Nagase Hongo Building 3F, 5-24-5  
Hongo,  
Bunkyo-ku, Tokyo, Japan

<sup>2</sup> Waseda University  
1-6-1 Nishi-Waseda, Shinjuku-ku,  
Tokyo, Japan

---

## Abstract

Scientific posters are used to present the contributions of scientific papers effectively in a graphical format. However, creating a well-designed poster that efficiently summarizes the core of a paper is both labor-intensive and time-consuming. A system that can automatically generate well-designed posters from scientific papers would reduce the workload of authors and help readers understand the outline of the paper visually. Despite the demand for poster generation systems, only a limited research has been conducted due to the lack of publicly available datasets. Thus, in this study, we built the SciPostLayout dataset, which consists of 7,855 scientific posters and manual layout annotations for layout analysis and generation. SciPostLayout also contains 100 scientific papers paired with the posters. All of the posters and papers in our dataset are under the CC-BY license and are publicly available. As benchmark tests for the collected dataset, we conducted experiments for layout analysis and generation utilizing existing computer vision models and found that both layout analysis and generation of posters using SciPostLayout are more challenging than with scientific papers. We also conducted experiments on generating layouts from scientific papers to demonstrate the potential of utilizing LLM as a scientific poster generation system. The dataset is publicly available at [https://huggingface.co/datasets/omron-sinicx/scipostlayout\\_v2](https://huggingface.co/datasets/omron-sinicx/scipostlayout_v2). The code is also publicly available at <https://github.com/omron-sinicx/scipostlayout>.

## 1 Introduction

Scientific posters are used to efficiently present the contributions of a scientific paper in a graphical format. A well-designed scientific poster conveys the essential elements of the research and requires less time to read than a paper. Unfortunately, creating a scientific poster that efficiently summarizes a paper is both labor-intensive and time-consuming. Although automating this task by utilizing ML models has shown promise, research on scientific poster

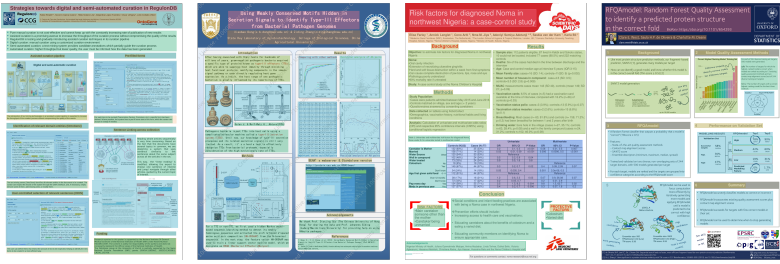


Figure 1: Example posters and annotations in SciPostLayout.

generation remains scarce due to the high complexity and multimodality of the task. Previous studies [29, 34] have built datasets to evaluate poster generation systems for scientific papers, but these datasets are either not publicly available or the data license is unclear, leaving this research field lacking in gold-standard benchmarks.

Previous studies [5, 58] have published publicly available datasets for layout analysis [8, 12, 23, 38] and layout generation [9, 11, 17, 35, 57]. However, these datasets focus on scientific papers or mobile application design, and neglect to examine the layout of scientific poster. Layout analysis and layout generation are important tasks for identifying the characteristics of effective layout design in a particular domain. Developing layout analysis models would be helpful in terms of identifying layout patterns common to well-designed posters. In addition, by developing layout generation models, we can generate layouts that follow the revealed patterns of attractive posters. In other words, building a layout dataset of scientific posters that can be utilized for these tasks would contribute significantly to the research on scientific poster generation.

In this paper, we introduce SciPostLayout, the first scientific poster layout dataset for layout analysis and generation. This dataset includes 7,855 scientific posters with manual layout annotations. All posters included in the dataset are under the CC-BY license<sup>1</sup>. Figure 1 shows examples of posters and annotations of SciPostLayout. SciPostLayout can be used to evaluate layout analysis and generation systems in the same way as existing datasets. Both the layout analysis and generation on SciPostLayout are more challenging than scientific papers because of the various positions of elements such as figures and tables. In addition, we manually collected 100 papers associated with posters in the dataset to utilize SciPostLayout in layout generation from scientific papers. These papers are also under the CC-BY license.

We evaluated existing models for layout analysis and generation on SciPostLayout. In the layout analysis task, we found that although the existing models can recognize certain elements with high accuracy, they are less effective than on the scientific paper dataset, indicating that layout analysis in SciPostLayout is more complicated. In the layout generation task, an LLM-based model can generate aligned layouts with less overlap, but we found that layout generation on SciPostLayout is also a more challenging task compared to generation on the scientific paper dataset. In addition, we implemented LLM-based models for generating poster layouts from papers and discussed the potential of poster generation utilizing LLM.

<sup>1</sup><https://creativecommons.org/licenses/by/4.0/>. Note that some of the older posters are under older licenses, such as CC-BY 2.0 and CC-BY 3.0.

## 2 Related Work

In this section, we describe related work on layout analysis, layout generation, and scientific poster generation.

### 2.1 Layout Analysis

Layout analysis is a task in which a system detects the layouts of unstructured documents by predicting bounding boxes and categories such as tables or figures. This task is generally categorized as an object detection problem [8, 23, 53].

The PubLayNet dataset [53] is one of the well-known layout analysis datasets. It contains scientific paper images automatically annotated with bounding boxes and polygonal segmentation across five categories: text, title, list, figure, and table. Previous studies have demonstrated that deep neural networks, including UDoc [8], DiT [23], and LayoutLMv3 [12], which are trained on the PubLayNet dataset, can recognize the layouts of scientific papers with high accuracy. These models utilize Faster R-CNN [30], Mask R-CNN [10], and Cascade R-CNN [9] as object detection models and ResNet [10] and Vision Transformer [6] as visual backbones.

Other studies have also proposed layout analysis datasets, such as Article Regions [32], which is a dataset of region-annotated scientific papers from PubMed Central, TableBank [24], an image-based table detection dataset built with Word and LaTeX documents, and DocBank [25], a document-level dataset with token annotations.

In contrast to the previous datasets, SciPostLayout focuses on scientific posters. The layout analysis of scientific posters is more challenging than that of scientific papers because of the variety of fonts and positions of figures and tables.

### 2.2 Layout Generation

Layout generation is a task that arose from the needs of design applications, including magazine covers, UI interfaces, presentation slides, and banner advertising [6, 7, 17, 35, 37]. Layout generation can be categorized into unconditional generation [10, 9, 15], which generates layouts without any constraints from the user, and conditional generation, which enables the user to create their own desired layout with constraints. There are also several subcategories within conditional generation, each giving the system different constraints for generation, such element types [18, 19, 20], element types and sizes [19], relationships between elements [18, 20], completion [9], and refinement [50].

Early studies explored generative adversarial networks (GANs) [18, 21] and variational autoencoders (VAEs) [10, 15, 55] for layout generation. Motivated by the success of the Transformer architecture [33] in NLP, some studies treated layout generation as a sequence-to-sequence problem [9, 16, 19, 26]. Moreover, discrete diffusion models have achieved notable results [4, 13, 12, 56]. Datasets including Rico [5], Magazine [37], and PubLayNet [53] are typically utilized to evaluate layout generation models.

We evaluate existing models for layout generation in the scientific poster domain. Generating layouts for scientific posters is challenging, as are other layout generation tasks that require the positioning of figures and tables.

## 2.3 Scientific Poster Generation

As stated earlier, scientific poster generation is a task in which a system generates a poster summarizing the important information in a scientific paper. Although this task is promising for applications, research on scientific poster generation remains scarce due to its high complexity and multimodality.

Paramita and Khodra [28] treated scientific poster generation as a summarization problem and developed a model to extract essential sentences from papers by classification and fill them into templates to generate posters. However, the generated posters are composed only of texts, making them unsuitable for efficiently conveying information.

Qiang et al. [29] decomposed poster generation into three subtasks: content extraction, panel attribute inference, and panel layout generation. For content extraction, their system uses TextRank [7] to summarize each section of a paper and then manually pick the important figures and tables. For panel attributes inference, the system uses a Bayesian network trained to learn the size and aspect ratio of each panel in a poster. For panel layout generation, the system generates panel layouts with a binary tree structure recursively. Finally, with the inferred data, posters are generated in a LaTeX format. The authors also constructed a dataset with 85 paper-poster pairs and facilitated evaluations of the model performances by experts on three metrics: readability, informativeness, and aesthetics. However, the licensing status of the posters is unclear, and the layout annotation is not detailed enough; for example, both figures and tables are simply labeled as *Figure*.

Xu and Wan [34] focused on the content extraction part of poster generation and proposed a model to extract text, figures, and tables from papers simultaneously. They additionally proposed a three-step framework to generate posters: section filter, content extraction, and poster composition. They also built a dataset for evaluation, which is currently awaiting public release.

Our SciPostLayout dataset is the first to contain scientific posters and papers with the CC-BY license, which can be used to evaluate scientific poster generation systems.

## 3 Dataset

Previous studies [29, 34] have built datasets to evaluate poster generation systems for scientific papers. However, these datasets are either not publicly available or the data licensing is unclear. In this paper, we construct SciPostLayout, a fully public dataset with all data under the Creative Commons license, which allows unrestricted dissemination, adaptation, and re-use.

First, we downloaded posters in PDF format from F1000Research<sup>2</sup>. Among these, we retained 7,943 posters under the CC-BY license. Posters under non-distributable and noncommercial licenses were excluded. The PDF files were converted into PNG format at DPI=100 for the following annotations. We excluded posters with file sizes below 200KB as they mainly consisted of text, which is unsuitable for layout analysis. After this exclusion, 7,855 posters remained. We investigated the word trends in the titles and found that most of the collected posters were in the biomedical field.

Next, we recruited professional data annotators to manually annotate the document layout of the 7,855 posters. The layouts of scientific posters are more diverse than the layouts of the papers in PubLayNet [38] because of the variety of fonts, size and position of figures

---

<sup>2</sup><https://f1000research.com/browse/posters>

Category	Contents
Title	paper title
Author Info	author, author affiliation
Section	section title <sup>a</sup>
Text	paragraph <sup>b</sup>
List	nested list <sup>c</sup>
Table	main body of table
Figure	main body of figure <sup>d</sup>
Caption	caption of table and figure
Unknown	advertising information, logo of affiliation <sup>e</sup>

<sup>a</sup>Only the highest level of sections are annotated as Section. Subsections are annotated as Text because of the small font size.

<sup>b</sup>When multiple paragraphs are connected, they are annotated as a single object. When a subsection is inserted between paragraphs, a new bounding box is created. The subsection name is not enclosed in a separate bounding box, but is included together with the subsequent paragraph. Footnotes are ignored during annotation.

<sup>c</sup>Not only bullet points (·) but also paragraph numbers (e.g. 1., a.) are treated as lists. It includes reference block.

<sup>d</sup>When sub-figures exist, the whole figure panel is annotated as a single object.

<sup>e</sup>Only elements with big areas are annotated.

Table 1: Annotation criteria of SciPostLayout.

Split	Title	Author Info	Section	Text	List	Table	Figure	Caption	Unknown	Total
Train	6,847	6,709	36,083	46,297	20,723	4,916	33,665	13,065	8	168,313
Dev	497	492	2,609	3,215	1,489	401	2,421	1,015	2	12,141
Test	498	494	2,646	3,199	1,583	328	2,484	1,010	1	12,243

Table 2: Statistics of train, dev, and test data in SciPostLayout.

and tables, and typography. We expanded PubLayNet’s five-category annotation standard to nine categories to acquire fine-grained annotations of the layouts. The annotation criteria are listed in Table 1.

Subsequently, we manually searched for the papers paired with the posters because there is no information relating to the papers on F1000Research. We found 100 papers under the CC-BY license and then divided these 100 paper-poster pairs into dev/test sets comprising 50 pairs each.

After the above processes, we obtained our dataset SciPostLayout, which includes train data with 6,855 posters and manual layout annotations as well as dev/test data with 500 posters, 50 papers paired with partial posters, and manual layout annotations. Table 2 lists the statistics of the dataset. SciPostLayout is the first scientific poster layout dataset that can be used for evaluating layout analysis and layout generation systems. Moreover, SciPostLayout is the first dataset that contains scientific paper-poster pairs with the CC-BY license, which can be utilized for evaluating scientific poster generation systems.

## 4 Experiments

We conducted three experiments using the collected dataset: layout analysis, layout generation, and paper-to-layout. We compared existing models for each experiment.

Model	Title	Author Info	Section	Text	List	Table	Figure	Caption	Overall
LayoutLMv3	<b>88.32</b>	<b>85.79</b>	<b>71.98</b>	<b>71.33</b>	<b>73.71</b>	<b>74.41</b>	<b>67.60</b>	<b>69.24</b>	<b>66.93</b>
DiT	85.23	79.89	70.64	67.66	69.93	74.16	67.12	59.27	63.77

Table 3: Layout analysis performance (mAP@IoU[0.50:0.95]) on SciPostLayout test set. **Bold** numbers indicate highest performances.

## 4.1 Layout Analysis

We used LayoutLMv3 [12] and DiT [23] for the layout analysis task. We started from the base size checkpoints with Cascade R-CNN detectors and fine-tuned them on SciPostLayout train data for both models. The checkpoint with the highest performances on the dev set was used for evaluation.

We measured the performance using the mean average precision (mAP) @ intersection over union (IoU) [0.50:0.95] of bounding boxes, the results of which are reported in Table 3. The *Unknown* category was omitted due to an insufficient number of elements. LayoutLMv3 outperformed DiT in all categories. In addition, both models showed high performances in the *Title* and *Author Info* categories. We attribute this result to the regularity of the *Title* and *Author Info* blocks, since they are always at the top of the posters and there is usually only one of each block per poster. However, compared to the results on PubLayNet [12, 23], in which mAP@IoU is over 90, both models showed a performance drop, indicating the complexity of our dataset.

## 4.2 Layout Generation

We conducted layout generation experiments with various settings for the information to be input into the model [16]. Generation conditioned on types (Gen-T) aims to generate layouts from the number of element types (categories). Generation conditioned on types and sizes (Gen-TS) aims to generate layouts from the number and size of element types. Generation conditioned on relationships (Gen-R) aims to generate layouts from the number of element types and position relationships between the elements. Completion means generating a complete layout from a part of the layout. Refinement means generating a new layout from a layout that needs improvement.

We used LayoutDM [14], LayoutFormer++ [16], and LayoutPrompter [26] for the layout generation task. Unlike the layout analysis models, the layout generation models were trained from randomly initialized parameters. We did not train LayoutPrompter because it uses GPT-4<sup>3</sup> via API.

We evaluated the model performances using maximum IoU (mIoU), Alignment, Overlap, and Fréchet Inception Distance (FID). mIoU is a measure of the highest IoU between a generated layout and a real layout [18, 26]. Alignment indicates how well the elements in a layout are aligned with each other [22, 26]. Overlap is the overlapping area between two arbitrary elements in a layout [22, 26]. FID measures how similar the distribution of the generated layouts is to that of real layouts [18, 26]. A higher mIoU value means higher performance; for the other metrics, a lower value means higher performance. Note that both mIoU and FID are evaluation metrics based on similarity to the real layouts, but mIoU is calculated from the intersection between layouts, while FID is an embedding-based evaluation metric. Thus, the hierarchical order of mIoU and FID performances across models is not always consistent.

<sup>3</sup>gpt-4-1106-preview

Task	Model	mIoU $\uparrow$	Alignment $\downarrow$	Overlap $\downarrow$	FID $\downarrow$
Gen-T	LayoutDM	0.073	0.002	0.524	<b>1.524</b>
	LayoutFormer++	0.079	0.002	0.242	26.999
	LayoutPrompter	<b>0.084</b>	<b>0.000</b>	<b>0.231</b>	9.470
Gen-TS	LayoutDM	<b>0.113</b>	0.002	0.564	<b>1.419</b>
	LayoutFormer++	0.089	0.003	0.245	27.372
	LayoutPrompter	0.102	<b>0.000</b>	<b>0.182</b>	8.605
Gen-R	LayoutDM	<b>0.075</b>	0.001	0.571	<b>5.556</b>
	LayoutFormer++	0.066	0.002	0.553	20.296
	LayoutPrompter	0.069	<b>0.000</b>	<b>0.317</b>	6.282
Completion	LayoutDM	–	0.001	0.610	<b>2.490</b>
	LayoutFormer++	–	0.001	0.250	5.840
	LayoutPrompter	–	<b>0.000</b>	<b>0.011</b>	10.874
Refinement	LayoutDM	0.119	0.002	0.531	1.196
	LayoutFormer++	0.301	0.002	0.289	10.955
	LayoutPrompter	<b>0.552</b>	<b>0.001</b>	<b>0.112</b>	<b>0.152</b>

Table 4: Layout generation performance on SciPostLayout test set. **Bold** numbers indicate the highest performances.  $\uparrow$  indicates larger values are ideal,  $\downarrow$  indicates smaller values are ideal. mIoU is omitted from the Completion setting because the number of elements of generated layouts differs from that of real layouts.

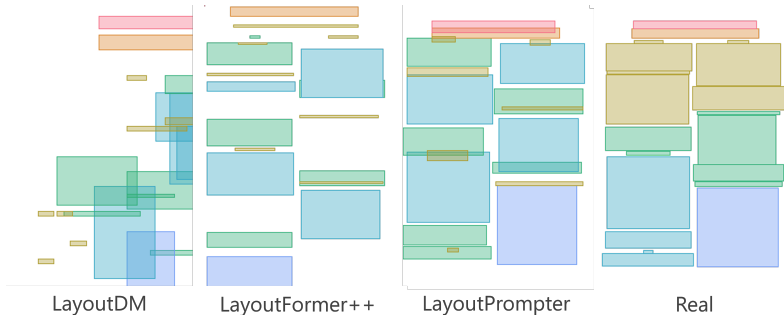


Figure 2: Examples of generated layouts and the real layout in the Refinement setting. Examples of the other settings are provided in Appendix A.

The results are shown in Table 4. mIoU was low for all models and was less than half that in PubLayNet [14, 16, 26]. In contrast, all models performed effectively on Alignment, indicating that they can generate aligned layouts. LayoutPrompter was the most effective for Overlap, indicating that it generates layouts with the least overlap. LayoutDM was the most effective in terms of FID, indicating that it can generate layouts most similar to real layout distribution on each setting. Comparing the FID and Overlap of LayoutDM and LayoutPrompter shows that LayoutDM tends to generate layouts that are similar to the real layout distribution but with more overlap, while LayoutPrompter generates layouts with less overlap but that are farther from the real layout distribution. In the Refinement setting, LayoutPrompter outperformed the other models, indicating that it can generate layouts similar to real layouts from noisy layouts. Figure 2 shows examples of generated layouts of each model and the real layout in the Refinement setting. LayoutPrompter generated the most similar layout to the real layout, while LayoutDM and LayoutFormer++ generated layouts with overlap that were not similar to the real layout.

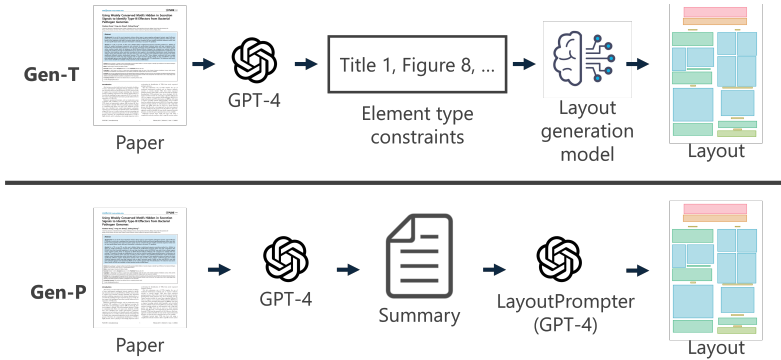


Figure 3: Gen-T and Gen-P settings of Paper-to-Layout.

Model	Title	Author Info	Section	Text	List	Table	Figure	Caption	Overall
GPT-4	0.06	0.04	1.90	4.9	2.31	0.57	3.66	2.43	1.98

Table 5: MAE results of element type extraction by GPT-4.

### 4.3 Paper-to-Layout

In Section 4.2, we conducted experiments on layout generation from real constraints. To develop a system that automatically generates a poster from a scientific paper, we need to implement a model that extracts constraints or generates a layout from a paper. Thus, we implement and compare models to generate layouts from scientific papers in two settings: Gen-T and Gen-P. Figure 3 illustrates the settings of Gen-T and Gen-P. In the Gen-T setting, GPT-4 first extracts element type constraints from a scientific paper. Then, a layout generation model generates a layout from the extracted constraints, the same as the Gen-T setting in Section 4.2. In the Gen-P setting, GPT-4 first generates a summary within 1,000 words from a scientific paper. Then, LayoutPromter with GPT-4 generates a layout from the paper summary. LayoutPromter does not directly generate a layout from a paper because the entire paper is too long for document retrieval when it searches for examples similar to the target paper for layout generation, which are given as few-shot examples. We used 50 paper-poster pairs of dev set as few-shot examples in the Gen-P setting. Different from Section 4.1 and 4.2, we used only 50 paper-poster pairs of test set to evaluate the model performances. In both settings, Nougat [2] was utilized to extract texts from a paper in PDF format. The prompts to extract element type constraints or generate a paper summary are shown in Appendix B.

We evaluated the accuracy of extracting element type constraints in the Gen-T setting based on mean absolute error (MAE). MAE is calculated as the mean of the absolute differences between the number of elements predicted by GPT-4 and the actual number of elements in the poster layout. For instance, if GPT-4 predicts 4 *Figures* while the actual poster has 5, the absolute error for this element type would be  $|4-5| = 1$ . We performed extraction three times and calculated the mean value. Table 5 shows the results. For *Title* and *Author Info*, the generated number of elements is close to the real constraints. For *Text* and *Figure*, GPT-4 shows large error values because the number of these elements is larger than the number of other elements and there is also a large range of values, as shown in Table 2. However, the overall MAE is less than 2, indicating that GPT-4 can generate constraints close to the real number of elements.



Task	Model	Alignment↓	Overlap↓	FID↓
Gen-T	LayoutDM	0.001	0.542	<b>9.073</b>
	LayoutFormer++	0.001	0.617	56.396
	LayoutPrompter	<b>0.000</b>	0.256	19.892
Gen-P	LayoutPrompter	<b>0.000</b>	<b>0.022</b>	17.790

Table 6: Paper-to-Layout generation performance on SciPostLayout test set. **Bold** numbers indicate the highest performances. ↓ indicates smaller values are ideal. mIoU is omitted because the number of elements of generated layouts differs from that of real layouts.

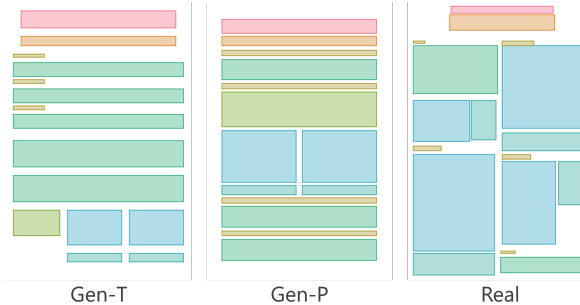


Figure 4: Example of LayoutPrompter generated layout and real layout in the Gen-T and Gen-P setting from scientific paper.

We conducted experiments on poster layout generation from scientific papers in the Gen-T and Gen-P settings, which are reported in Table 6. The number of each element of the extracted constraints in the Gen-T setting was given to the models as the mean value rounded down to the nearest whole number. All models performed effectively on Alignment, indicating that they can generate aligned layouts in the same way as layout generation from real constraints in Section 4.2. LayoutPrompter in the Gen-P setting was the most effective for Overlap, indicating that it generates layouts with the least overlap. Comparing the FID values in the Gen-T setting to the values in the Gen-T setting of Table 4, we find that the FID values of all models slightly increase, indicating that generating layouts similar to real layouts from extracted constraints is more difficult than generating layouts from real constraints. Figure 4 shows examples of layouts generated by LayoutPrompter in the Gen-T and Gen-P settings. Although the generated layouts are not similar to the real layout, LayoutPrompter could generate aligned layouts with less overlap. The experimental results in this section demonstrate that although the current LLM has difficulty generating layouts similar to real layouts from scientific papers, it can generate high-quality layouts, indicating that LLM has potential as a poster generation system from scientific papers.

## 5 Conclusion

In this paper, we built a new dataset called SciPostLayout, which consists of 7,855 scientific posters and 100 scientific papers downloaded from a website. All posters in SciPostLayout are manually annotated with categories of layout elements (such as titles and figures) to be used in layout analysis and generation tasks. In addition, SciPostLayout is available for commercial research because all of the posters and papers are under the CC-BY license. We conducted layout analysis and generation experiments to evaluate the performances of

existing models on SciPostLayout. For layout analysis, although some elements could be recognized with high accuracy, we found that layout analysis on SciPostLayout was more difficult than on a scientific paper dataset. For layout generation, although existing models could generate aligned layouts, we found it was difficult to generate layouts that are similar to real layouts. In addition, we implemented and evaluated GPT-4-based models to generate poster layouts from scientific papers. The experimental results proved that LLM has potential as a scientific poster generation system. Our future work will involve developing a model to improve layout analysis and generation. In addition, we will investigate improved methods for generating poster layouts and extracting contents from scientific papers.

## Acknowledgements

This work was supported by JST Moonshot R&D Program, Grant Number JPMJMS2236. We would like to thank Naoto Inoue, Ph. D. of CyberAgent Inc. AILab for useful comments about layout generation models.

## References

- [1] Diego Martin Arroyo, Janis Postels, and Federico Tombari. Variational transformer networks for layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13642–13652, 2021.
- [2] Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural Optical Understanding for Academic Documents, 2023.
- [3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.
- [4] Shang Chai, Liansheng Zhuang, and Fengying Yan. LayoutDM: Transformer-based Diffusion Model for Layout Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18349–18358, 2023.
- [5] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibsichman, Daniel Afegan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pages 845–854, 2017.
- [6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [7] Tsu-Jui Fu, William Yang Wang, Daniel McDuff, and Yale Song. DOC2PPT: automatic presentation slides generation from scientific documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 634–642, 2022.

- [8] Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Nikolaos Barmaliou, Ani Nenkova, and Tong Sun. Unidoc: Unified pretraining framework for document understanding. *Advances in Neural Information Processing Systems*, 34:39–50, 2021.
- [9] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. LayoutTransformer: Layout Generation and Completion with Self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1004–1014, 2021.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [12] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.
- [13] Mude Hui, Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, Yuwang Wang, and Yan Lu. Unifying Layout Generation with a Decoupled Diffusion Model. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1942–1951, 2023.
- [14] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. LayoutDM: Discrete Diffusion Model for Controllable Layout Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10167–10176, 2023.
- [15] Zhaoyun Jiang, Shizhao Sun, Jihua Zhu, Jian-Guang Lou, and Dongmei Zhang. Coarse-to-fine generative modeling for graphic layouts. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 1096–1103, 2022.
- [16] Zhaoyun Jiang, Jiaqi Guo, Shizhao Sun, Huayu Deng, Zhongkai Wu, Vuksan Mijovic, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. LayoutFormer++: Conditional Graphic Layout Generation via Constraint Serialization and Decoding Space Restriction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18403–18412, June 2023.
- [17] K. Kikuchi, M. Otani, K. Yamaguchi, and E. Simo-Serra. Modeling Visual Containment for Web Page Layout Optimization. *Computer Graphics Forum*, 40(7):33–44, 2021.
- [18] 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. doi: 10.1145/3474085.3475497.

- [19] 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.
- [20] 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, *European Conference on Computer Vision*, pages 491–506, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58580-8.
- [21] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. Layoutgan: Generating graphic layouts with wireframe discriminators. *arXiv preprint arXiv:1901.06767*, 2019.
- [22] 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. doi: 10.1109/TVCG.2020.2999335.
- [23] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. DiT: Self-supervised Pre-training for Document Image Transformer. In *ACM Multimedia*, October 2022.
- [24] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. TableBank: A Benchmark Dataset for Table Detection and Recognition, 2019.
- [25] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. DocBank: A Benchmark Dataset for Document Layout Analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 949–960, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.82.
- [26] Jiawei Lin, Jiaqi Guo, Shizhao Sun, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. LayoutPrompter: Awaken the Design Ability of Large Language Models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [27] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Text. In Dekang Lin and Dekai Wu, editors, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [28] Kanya Paramita and Masayu Leylia Khodra. Tailored Summary for automatic poster generator. In *International Conference On Advanced Informatics: Concepts, Theory And Application*, pages 1–6, 2016. doi: 10.1109/ICAICTA.2016.7803112.
- [29] Yu-Ting Qiang, Yan-Wei Fu, Xiao Yu, Yan-Wen Guo, Zhi-Hua Zhou, and Leonid Sigal. Learning to Generate Posters of Scientific Papers by Probabilistic Graphical Models. *Journal of Computer Science and Technology*, 34(1):155–169, 2019. ISSN 1000-9000(Print) /1860-4749(Online). doi: 10.1007/s11390-019-1904-1.

- [30] Soliha Rahman, Vinoth Pandian Sermuga Pandian, and Matthias Jarke. Ruite: Refining ui layout aesthetics using transformer encoder. In *26th International Conference on Intelligent User Interfaces-Companion*, pages 81–83, 2021.
- [31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.
- [32] Carlos Soto and Shinjae Yoo. Visual Detection with Context for Document Layout Analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3464–3470, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1348.
- [33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [34] Sheng Xu and Xiaojun Wan. Neural Content Extraction for Poster Generation of Scientific Papers, 2021.
- [35] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5481–5489, 2021.
- [36] Junyi Zhang, Jiaqi Guo, Shizhao Sun, Jian-Guang Lou, and Dongmei Zhang. Layout-Diffusion: Improving Graphic Layout Generation by Discrete Diffusion Probabilistic Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.
- [37] Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson WH Lau. Content-aware generative modeling of graphic design layouts. *ACM Transactions on Graphics*, 38(4):1–15, 2019.
- [38] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. PubLayNet: largest dataset ever for document layout analysis. In *International Conference on Document Analysis and Recognition*, pages 1015–1022. IEEE, Sep. 2019. doi: 10.1109/ICDAR.2019.00166.