DocSynthv2: A Practical Autoregressive Modeling for Document Generation

Sanket Biswas[1*] Rajiv Jain[2] Vlad I. Morariu[2] Jiuxiang Gu[2] Puneet Mathur[2]
Curtis Wigington[2] Tong Sun[2] Josep Lladós[1]

[1] Computer Vision Center, UAB, Spain [2] Adobe Research
{sbiswas, josep}@cvc.uab.es {rajijain, morariu, jigu, puneetm, wigingto, tsun}@adobe.com

Abstract

While the generation of document layouts has been extensively explored, comprehensive document generation—encompassing both layout and content—presents a more complex challenge. This paper delves into this advanced domain, proposing a novel approach called ***DocSynthv2*** *through the development of a simple yet effective autoregressive structured model. Our model, distinct in its integration of both layout and textual cues, marks a step beyond existing layout-generation approaches. By focusing on the relationship between the structural elements and the textual content within documents, we aim to generate cohesive and contextually relevant documents without any reliance on visual components. Through experimental studies on our curated benchmark for the new task, we demonstrate the ability of our model combining layout and textual information in enhancing the generation quality and relevance of documents, opening new pathways for research in document creation and automated design. Our findings emphasize the effectiveness of autoregressive models in handling complex document generation tasks.*

1. Introduction

Recent advancements in generative models [3, 5, 31, 33] have made significant impacts on language, image, and multimodal content generation. There is an increasing focus on vector graphic document generation [11, 24, 29] within this realm, where these models support users in creating, modifying, publishing, and designing both business and artistic documents. Documents differ from standard natural images as they contain structured layers of text and media content. The field of document generation presents unique challenges in seamlessly integrating visual elements such as style, layout, and multimedia with textual content, posing new problems for the vision community.

Document layout generation [1, 7, 9, 10, 14, 16, 19] has played a crucial role in numerous applications, ranging from automated report creation to dynamic webpage design, significantly impacting how information is perceived and interacted with by users. With large language models (LLMs) [3, 27] becoming more and more capable of compositional reasoning of visual concepts [6], it opens further avenues for exploiting autoregressive approaches in the automatic end-to-end generation of both document content and layout structure. Moreover, synthetic document generation [2, 22] has gained attention in recent times owing to lack of multi-domain large-scaled layout annotated datasets necessary for document pre-training [15]. However, end-to-end pixel-based approaches [2, 30] suffer from low-resolution generated outputs where the textual content can be hardly extracted. In this work, we introduce **DocSynthv2** to seamlessly generate layout structure with integrated text, essential to convey specific information and context, completing the communication objective of the document.

This work contributes to document generation research in three different folds: 1) We curate a large-scale extended benchmark called **PubGenNet** tailor-made for document generation and completion task. 2) We introduce a simple and flexible autoregressive approach for generating high-resolution document outputs, capable of handling sequences of arbitrary lengths. 3) We outline future challenges and opportunities in evaluating document generation, setting the stage for advancements in this evolving field.

2. Related Work

Document Layout Generation Recently, there has been a surge in research on layout generation. Foundational works like LayoutGAN [18] and LayoutVAE [14] have been influential in synthesizing layouts by modeling geometric relations of different 2D elements and then rendering them in the image space. Document layout generation has received extensive interest in recent years owing to its integration in tasks such as content generation [29, 34] and graphic web designs [4]. While [34] attempts to generate document layouts

[*] Work done during internship at Adobe Research

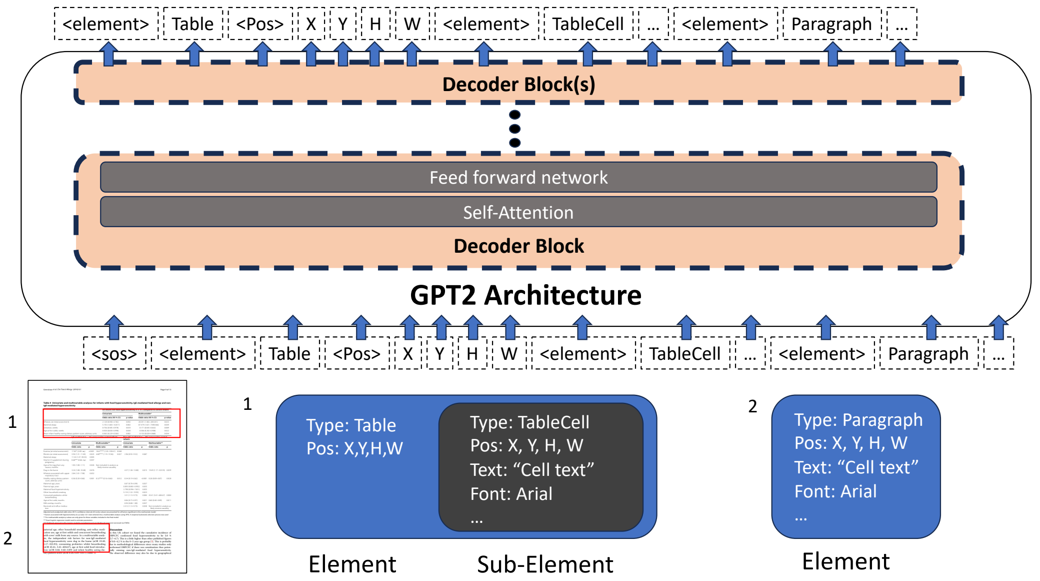


Figure 1. **Overall Architecture of `DocSynthv2`.**

given user-conditioned prompts (eg. input reference image, keywords, and category of the document), [19] proposed an approach to construct hierarchies of document layouts and later sample and generate them using a recursive VAE. The aforementioned method was further extended using graph autoencoder networks [20] with optional design constraints for further improvement. Gupta *et. al.* [7] proposed `Layout Transformer` with self-attention [28] which is the most relevant to this work. They used a next element prediction objective (i.e. layout completion) using a transformer architecture in an autoregressive manner to produce layout tokens, including class labels and bounding boxes of document objects. [1] tried to combine these generative transformers [7] with VAE's to learn better layout refinement and prediction.

Synthetic Document Generation The Computer Vision community has also captured emerging interest to generate synthetic realistic scene images with plausible layouts from a user provided reference layout [8, 13, 32], emphasizing particularly on high-resolution image outputs. The DocSynth framework [2] introduced the first image-to-image translation pipeline for creating synthetic document image datasets for augmenting real data during training for document layout analysis tasks [21, 35]. In this work, we move a step forward towards generating synthetic data with content preservation.

3. Method

In this section, we introduce our proposed approach for the document generation task. We first discuss our representation of document elements essential for model understanding. Next, we discuss the DocSynthv2 framework and show how we can leverage the knowledge of both layout elements and their corresponding content to model the probability distribution of an overall page structure. Lastly, we discuss the learning objectives we have used to train the whole network.

3.1. Overview

Document Representation The document layout of a page can comprise multiple sets of elements, where each element can be described by its category c, left and top coordinate x and y, as well as width w and height h. The continuous attributes x, y, w and h are often quantized, which has proven to be useful for graphic layout generation approaches [1, 7, 34]. Following the FlexDM approach [11], we represent document $\mathcal{D}$ as a vector consisting of a tuple of layout components $(D_1, D_2, \dots, D_S)$, where S is the number of elements in $\mathcal{D}$. Each element $D_i = \{d_i^k \mid k \in \mathcal{E}\}$ can represent either element type , position, style attributes, or raw text content where k represents the indices of the attributes. Contrary to FlexDM [11], we do not use any embeddings in the input sequence but rather use only the element's layout information or its content attributes. We concatenate the layout information along with the text attribute tokens for every element as shown in Equation 1. Here, N represents the total number of elements, while $\langle\text{sos}\rangle$ and $\langle$ eos $\rangle$ are special tokens which denote the start and end of a sequence. Also, a special `[NULL]` token appears when d_i^k is inevitably missing (e.g., font type for a non-text element), or padding variable-length sequences when training a mini-batch.

$$\mathcal{D} = \{\langle\text{sos}\rangle c_1 x_1 y_1 w_1 h_1 t_1 \dots c_N x_N y_N w_N h_N t_N \langle\text{eos}\rangle\} \tag{1}$$

Representing with Discrete Variables Following Layout-Transformer [7], we applied an 8-bit uniform quantization on every document element (image region or text) and modelled them using Categorical distribution. We note that while converting coordinates into discrete values leads to some loss of precision, this approach enables the modeling of multiple kinds of distributions, which is crucial for document layouts. Every document object (text or non-text) is projected to the same dimension such that we can concatenate every element $(c_N, x_N, y_N, w_N, h_N, t_N)$ in a single linear sequence of their element values. The overall structure of a page can then be represented by a sequence of m latent vectors where m is decided by the total number of tokens encoded in the input sequence S. For conciseness, we use $\boldsymbol{\theta}_j, j \in \{1, \ldots, m\}$ to represent any document element in the above sequence. We model this joint distribution as a product over a series of conditional distributions using the chain rule as shown in Equation 2.

$$p\left(\boldsymbol{\theta}_{1:m}\right) = \prod_{j=1}^{m} p\left(\boldsymbol{\theta}_j \mid \boldsymbol{\theta}_{1:j-1}\right) \tag{2}$$

3.2. Model Architecture

DocSynthv2 is a document generation transformer pretrained on document datasets containing multiple elements with a combined set of layout and text attributes. The model learns neural representations of document data, capturing both physical and logical relationships of the document elements with the previously predicted element. Our overall architecture of DocSynthv2 is shown in Figure 1.

Training: Given an initial set of T visible tokens as input containing attributes representing: 1) Layout Category (eg. Table, Table Cell, Paragraph, Title, Caption etc.) 2) Position 3) Font Style 4) Text Content, the model tries to predict the next element with an autoregressive GPT-2 Transformer decoder [23]. Each of these GPT blocks consists of a masked multi-head attention (MHA) and a feedforward network (FFN) as shown. The output at the final layer corresponds to the next parameter.

Inference: During inference, both the position and text tokens are synthesized auto-regressively for the fixed category token (i.e. a reference layout you would like to generate). During both training and inference, the ground-truth sequences have been used to train the model more efficiently as done in [7].

Losses: Since the model has both continuous and discrete sets of parameters as already discussed, we use a variational loss to minimize KL-Divergence between the softmax predictions for all discrete parameters as in [7].

4. Experiments

4.1. Datasets

Our evaluation of DocSynthv2 primarily utilizes two vector graphic document datasets, Crello [29] and DocGenNet, our curated version of PubLayNet [35] streamlined for the task of document generation.

Crello: Originating from an online design platform, this dataset encompasses a broad range of design templates, including but not limited to social media posts, banner ads, blog headers, and printed materials. We use a similar experimental setting as used in FlexDM [11]. The released dataset by the authors was partitioned into 18,738 training instances, 2,313 for validation, and 2,271 for testing. Detailed definitions of each attribute can be found in the original paper [29].

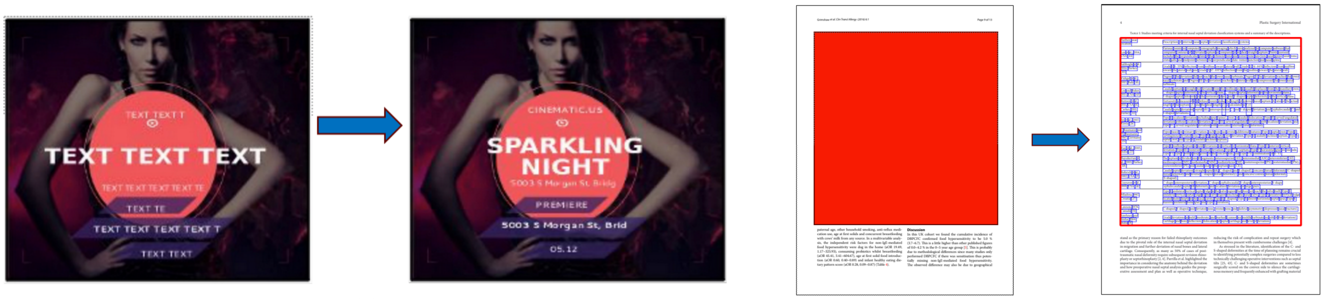
PubGenNet: For experimental validation, we generated a new benchmark called "PubGenNet," a large-scaled extended dataset curated to advance the field of document generation. This dataset was assembled by extracting a diverse array of samples from the original PubLayNet dataset [35], which itself is derived from an extensive collection of scientific publications available in PubMed Central. To ensure a comprehensive set of text attributes (eg. font type) along with raw textual content, we utilized a PDF extraction procedure, using the PyMuPDF library enabling us to align this extracted data with the original COCO annotations. In summary, the overall curation process involved extraction and processing of layout and text data from a set of documents represented in the PubLayNet format. After obtaining the document-specific attributes, the processed data was compiled into a structured dataset suitable for training and evaluating document generation models. The resulting training and validation instances are similar to the dataset statistics in PubLayNet with 335,703 document samples for training and 11,245 instances for validation.

4.2. Tasks

The primary motivations for our model are to address the key aspects of document design and generation. We have selected the evaluation tasks based on: (1) Creating a new document or completing a partially finished one, focusing on maintaining coherence, appearance, and relevance to the intended content. (2) Test the model's ability in layout design, specifically its understanding of spacing, alignment, and the interplay between text and other elements.

Document Completion: This task requires the model to analyze the current layout elements and content within the document (eg. text, title, tables, figures etc.) and logically predict what elements should follow to maintain the coherence and plausible structure of a document.

Single and Multiple Text Box Placement: This task in terms of next element prediction requires the model to identify optimal locations and sizes for text boxes within a document, based on the existing layout and design principles. It assesses the model's capability to seamlessly incorporate new text elements, ensuring they align with the document's structure and visual appeal.



(a)

(b)

Figure 2. Qualitative results over the two tasks a) **Text Prediction**: Input (L) and Predicted text content (R) for a Crello sample b) **Document Completion**: Input (L) with partially filled document and Predicted output (R) for PubGenNet sample

4.3. Quantitative Evaluation

Table 1 summarizes the performance comparison of DocSynthv2 over the existing SOTA transformer decoder-only models. Our full model (with text attributes) gives us boost in performance over the layout-only model, demonstrating that utilizng the raw text can help guide models for layout generation when avaialble. Although our model is a lightweight decoder-only architecture, it can perform on par with LayoutFormer++ [12] which is an encoder-decoder-based transformer. Our results with high Alignment and Overlap scores also suggest that *layout generation and completion models gain substantial improvement when trained on sequences integrating textual content*.

In Table 2, we summarize the performance of Single and Multiple Text Box Placement in the Crello dataset. The results show that the model does worse for text placement in the Single Text box condition, likely due to the weaker multimodal features compared to [11]. However, it performs on par for IoU and outperforms for BDE in the Multiple condition, which may be due to the raw text in our model.

	mIoU↑	FID↓	Align↓	Over↓
LayoutTrans [7]	0.077	14.769	0.019	**0.0013**
Layoutformer++ [12]	**0.471**	**10.251**	0.020	0.0022
Ours (w/o txt)	0.315	12.217	0.025	0.0019
Ours (lay+txt)	0.452	10.718	**0.015**	**0.0013**
Δ	-0.019	0.467	-0.004	0.00

Table 1. Quantitative evaluation for Document Completion. Results style: **best**, second best. ↑ higher is better and ↓ lower is better

4.4. Qualitative Evaluation

Figure 2 shows example of our applied for text synthesis and document completion on the Crello and PubGenNet datasets. In the Crello Text prediction example, it can be seen that the text is aligned with the layout showing a plausible flyer title for the heading section followed by an address and date in the sub text fields. For the Document Completion Task, we have the model generate the text within in an existing Table structure. The filled text maintains coherence across the two table columns, filling it with Authors names and reference information on the left and text of the right. In this example the text coherence could likely be improved by LLMs.

	Single		Multiple	
	IoU↑	BDE↓	IoU↑	BDE↓
SmartText [17]	0.047	0.262	0.023	0.300
FlexDM (MM) [11]	**0.357**	**0.098**	**0.110**	0.141
FlexDM (w/o img) [11]	0.355	0.100	0.103	0.156
FlexDM (w/o txt) [11]	0.350	0.106	0.086	0.178
Ours	0.315	0.104	0.105	**0.131**

Table 2. Quantitaive evaluation for Single and Multiple Box Placement in Crello. Results style: **best**, second best. ↑ higher is better and ↓ lower is better

5. Future Scope and Challenges

In conclusion, DocSynthv2 demonstrates that integrating text with layout sequences into an autoregressive framework enriches the data representation and provides additional context, leading to improved stability and performance in generating coherent and contextually appropriate document content and motivates future work. First, the integration of layout and text needs to advance beyond current capabilities to *address the diversity of document styles and industry-specific standards*. We believe future work may benefit from visual-language models [26] that can understand multimodal content or code generation models [25] that can learn complex structure from a wide array of document formats and content types. We also believe, the evaluation of document generation systems remains a critical challenge. There is a pressing need for *evaluation frameworks which can effectively measure the usefulness of generated documents* in terms of both their visual layout and textual content. These frameworks must encompass metrics that evaluate coherence, relevance, readability, and visual appeal, reflecting the multi-functional nature of documents.

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. 1, 2

[2] Sanket Biswas, Pau Riba, Josep Lladós, and Umapada Pal. Docsynth: a layout guided approach for controllable document image synthesis. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part III*, pages 555–568. Springer, 2021. 1, 2

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[4] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, 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. 1

[5] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 1

[6] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[7] 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. 1, 2, 3, 4

[8] Sen He, Wentong Liao, Michael Ying Yang, Yongxin Yang, Yi-Zhe Song, Bodo Rosenhahn, and Tao Xiang. Context-aware layout to image generation with enhanced object appearance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15049–15058, 2021. 2

[9] Mude Hui, Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, Yuwang Wang, and Yan Lu. Unifying layout generation with a decoupled diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1942–1951, 2023. 1

[10] 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. 1

[11] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Towards flexible multi-modal document models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14287–14296, 2023. 1, 2, 3, 4

[12] 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, 2023. 4

[13] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018. 2

[14] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9895–9904, 2019. 1

[15] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer, 2022. 1

[16] 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

[17] Chenhui Li, Peiying Zhang, and Changbo Wang. Harmonious textual layout generation over natural images via deep aesthetics learning. *IEEE Transactions on Multimedia*, 24: 3416–3428, 2021. 4

[18] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. Layoutgan: Generating graphic layouts with wireframe discriminators. *arXiv preprint arXiv:1901.06767*, 2019. 1

[19] Akshay Gadi Patil, Omri Ben-Eliezer, Or Perel, and Hadar Averbuch-Elor. Read: Recursive autoencoders for document layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 544–545, 2020. 1, 2

[20] Akshay Gadi Patil, Manyi Li, Matthew Fisher, Manolis Savva, and Hao Zhang. Layoutgmn: Neural graph matching for structural layout similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11048–11057, 2021. 2

[21] Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter Staar. Doclaynet: a large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3743–3751, 2022. 2

[22] Lorenzo Pisaneschi, Andrea Gemelli, and Simone Marinai. Automatic generation of scientific papers for data augmentation in document layout analysis. *Pattern Recognition Letters*, 167:38–44, 2023. 1

[23] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario

Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 3

[24] Wataru Shimoda, Daichi Haraguchi, Seiichi Uchida, and Kota Yamaguchi. Towards diverse and consistent typography generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7296–7305, 2024. 1

[25] Zecheng Tang, Chenfei Wu, Juntao Li, and Nan Duan. Layoutnuwa: Revealing the hidden layout expertise of large language models. *arXiv preprint arXiv:2309.09506*, 2023. 4

[26] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19254–19264, 2023. 4

[27] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[28] 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. 2

[29] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5481–5489, 2021. 1, 3

[30] Moonbin Yim, Yoonsik Kim, Han-Cheol Cho, and Sungrae Park. Synthtiger: Synthetic text image generator towards better text recognition models. In *International conference on document analysis and recognition*, pages 109–124. Springer, 2021. 1

[31] Junyi Zhang, Jiaqi Guo, Shizhao Sun, Jian-Guang Lou, and Dongmei Zhang. Layoutdiffusion: Improving graphic layout generation by discrete diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7226–7236, 2023. 1

[32] Bo Zhao, Weidong Yin, Lili Meng, and Leonid Sigal. Layout2image: Image generation from layout. *International Journal of Computer Vision*, 128(10):2418–2435, 2020. 2

[33] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22490–22499, 2023. 1

[34] Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson WH Lau. Content-aware generative modeling of graphic design layouts. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019. 1, 2

[35] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International conference on document analysis and recognition (ICDAR)*, pages 1015–1022. IEEE, 2019. 2, 3