

Sketch-to-Layout: Sketch-Guided Multimodal Layout Generation

Riccardo Brioschi^{*†} Aleksandr Alekseev^{*†} Emanuele Nevali^{*†} Berkay Döner^{*†}

Omar El Malki^{*†} Blagoj Mitrevski[‡] Leandro Kieliger[‡] Mark Collier[‡] Andrii Maksai[‡]

Jesse Berent[‡] Claudiu Cristian Musat[‡] Efi Kokiopoulou[‡]

Abstract

Graphic layout generation is a growing research area focusing on generating aesthetically pleasing layouts ranging from poster designs to documents. While recent research has explored ways to incorporate user constraints to guide the layout generation, these constraints often require complex specifications which reduce usability. We introduce an innovative approach exploiting user-provided sketches as intuitive constraints and we demonstrate empirically the effectiveness of this new guidance method, establishing the sketch-to-layout problem as a promising research direction, which is currently under-explored. To tackle the sketch-to-layout problem, we propose a multimodal transformer-based solution using the sketch and the content assets as inputs to produce high quality layouts. Since collecting sketch training data from human annotators to train our model is very costly, we introduce a novel and efficient method to synthetically generate training sketches at scale. We train and evaluate our model on three publicly available datasets: PubLayNet [41], DocLayNet [30] and SlidesVQA [33], demonstrating that it outperforms state-of-the-art constraint-based methods, while offering a more intuitive design experience. In order to facilitate future sketch-to-layout research, we release O(200k) synthetically-generated sketches for the public datasets above.¹

1. Introduction

Designing aesthetically pleasing and usable layouts for graphic design is a fundamental challenge. Layouts should

^{*}These authors contributed equally and are listed in a [certified random order](#). [†]EPFL, work done during time as Student Researchers at Google DeepMind. [‡]Google DeepMind. Correspondence to: Blagoj Mitrevski bmitrevski@google.com ¹The datasets are available at https://github.com/google-deepmind/sketch_to_layout

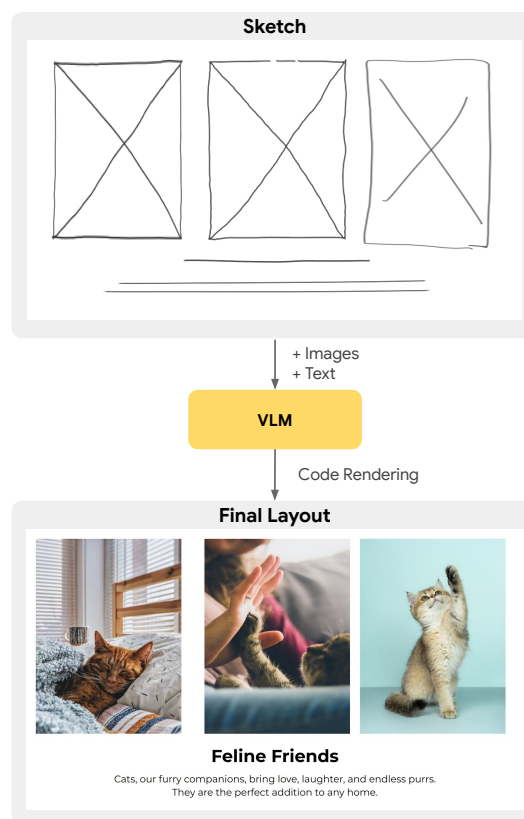


Figure 1. Our sketch-to-layout approach leverages sketches to guide the generation of multimodal layouts in a natural and intuitive way.

represent a visually pleasing arrangement of text and image elements with appropriate sizes and positions, while at the same time capturing the right information hierarchy. Assets should have consistent semantic relationships such as an engaging reading order. Manual design can be time-consuming and automated layout generation aims to reduce

this burden.

Recent research has explored various approaches to layout generation, including image generation methods such as GANs [18, 21] and LLM-based methods [26, 34, 39]. Some of these approaches try to incorporate user-defined constraints to guide the generation, but require complex details such as specifying precise element dimensions [16, 26], complex positional relationships [16, 18, 26, 39], grid-based guidelines [6] and detailed textual descriptions [25]. It is cumbersome for users to come up with such complex constraints. On the contrary, sketching, a common design practice where users quickly outline the structure of a design, offers a more intuitive alternative. Sketching is a widely used technique for creative tasks that captures the high-level essence of a layout without requiring overwhelming detail. Studies on designer behavior [28, 29] show that sketching is an integral starting point for almost all designers in various domains [4].

This paper proposes using sketches as intuitive constraints to guide the generation of multimodal (image and text) layouts. We start by demonstrating empirically the effectiveness of sketches as a new guidance method when compared against other forms of user-defined constraints. This empirical result suggests that the sketch-to-layout problem is a promising direction for constrained layout generation, which has been under-explored. To tackle the sketch-to-layout problem, we propose a solution leveraging Vision-Language Models (VLMs); see also Fig 1. Our method takes as input a user constraint in a form of a sketch along with image and text assets, to produce visually pleasing high quality layouts, capturing the structure suggested by the user while maintaining aesthetic appeal.

Although VLMs have shown impressive performances on a wide range of tasks, recent research [23] demonstrated that generating the correct layout from sketch inputs in a single step is challenging even for state-of-the-art VLMs, which amplifies the need of collecting sketch training data for improving performance. However, collecting sketch data from human annotators to train a VLM model is very costly and time-consuming. In order to address this challenge, we propose a novel and efficient technique to synthetically generate sketches at scale, unblocking the fine-tuning of VLMs to tackle the sketch-to-layout problem. In order to accelerate research progress on sketch-to-layout, we release a dataset of $O(200k)$ synthetic sketches generated by our proposed method.

Our VLM-based approach is general and applicable to any VLM. In our empirical study we use PaLiGemma 3B [3] as an example open-source VLM. We train and evaluate our model on three publicly available datasets: PubLayNet [41], DocLayNet [30] and SlidesVQA [33], demonstrating that it outperforms state-of-the-art constraint-based methods, while offering a more intuitive design experience.

By evaluating our model on both synthetic and human-produced sketches, we arrive at comparable performance, which validates the use of synthetic sketches as a reliable proxy for actual human-produced sketches when used as training data for VLMs in the sketch-to-layout task. In summary, our contributions are as follows:

- We demonstrate the value of sketches as a novel guidance method for layout generation, establishing the sketch-to-layout as an effective research direction for guided layout generation.
- We introduce a novel methodology to create large-scale synthetic datasets with sketches of documents and layouts, unblocking efficient VLM training and evaluation.
- We release our large collection of $O(200k)$ synthetically generated sketches for three publicly available datasets, in order to facilitate future research in this previously data-scarce domain.
- We provide experimental results showing that our method leveraging PaLiGemma outperforms state-of-the-art constraint-based layout generation methods by more than 40% in terms of Maximum IoU on three publicly available datasets. Our empirical results also highlight the importance of the content-awareness aspect of our method.
- We introduce Content Ordering Score (COS), a new metric inspired by the order loss [22], designed to assess the content-awareness of a generated layout.

2. Related Work

Unconstrained Generation. Early research in layout generation primarily focused on unconditional generation. CanvasVAE [38] models documents as a combination of canvases and elements, adopting a VAE to capture the distribution of their attributes. Gupta et al. [11] propose an auto-regressive transformer to frame layout generation as a sequence-to-sequence task, and show the effectiveness of their approach on different domains.

Constrained Generation. We focus on constrained generation of layouts, which has been investigated before with different types of constraints. LayoutVAE [17] proposes a two-stage VAE model that takes the set of labels as input constraint. Similarly, LayoutGAN [21] synthesizes layouts given the set of labels with each label having a separate probability distribution in the generator. Further work from Li et al. [22] uses the area, aspect ratio and reading-order of the input elements as the input constraints. [19] takes relational constraints between elements (such as specifying a text block to be on the left of an image) and models the relationships using a graph-based model. [18] uses a transformer-based GAN that can additionally take beautification constraints, such as alignment and non-overlap. Later work using diffusion models [6, 13], transformers [16], and LLMs [26] focus on respecting different con-

straints including layout grids, category types and sizes of elements and the relationships between them. Recent work incorporates textual descriptions of layouts to guide the generation [25, 26]. While textual descriptions are a step towards more intuitive constraints compared to the previously adopted ones, we argue that sketches offer an even more direct and natural way for users to express their layout preferences.

Content-Aware Methods. Some prior work has incorporated content in the form of user-provided assets to guide the generation. Xinru Zheng and Lau [37] take images, keywords and the layout category as input, which are fed into separate encoders guiding a layout generation GAN. CGL-GAN [42] takes the background canvas as input and leverages its saliency map to guide the generation process. Later work in the area also focused on placing elements on a provided background using different architectures, such as CNN-LSTM-based-GAN [12] and pretrained VLMs [31, 39]. Yu et al. [40] uses an object detection transformer model (DETR[5]) to guide the generation, relying on ViT[9] and BERT [8] to encode input images and texts. [14] considers documents as a set of multimodal elements and uses CLIP to embed textual and visual features. Similarly, [32] utilizes CLIP embeddings for input elements and vector attributes as conditions to diffusion model. Similar to this prior work, we let the user provide images and texts as input and use a VLM to encode these assets.

Sketch-based methods. Sketch as an input constraint has been mostly used in the GUI design literature, with sketches containing interface components similar to wireframes. Jain et al. [15] use a ResNet-based object detection model to convert sketches to JSON objects in real time. Similarly, Baulé et al. [2], Liang and Lin [24], Mohian and Csallner [27] use different architectures to solve this task as real-time object detection. Ferreira et al. [10] generate synthetic sketches using some heuristics and use them to pre-train the final model. Compared to these methods, our approach allows end-to-end generation with user-provided assets, tackling the broad graphical layout generation while previous methods focused on GUIs only.

Multimodal Transformer Models. Recent work has tried to apply large language models to the layout generation problem. Tang et al. [34] treats layout generation as a code generation problem, by converting layouts into SVG strings and using CodeLLaMA to solve the problem. Lin et al. [26] use a few-shot prompted GPT, dynamically selecting the in-context examples to be included in the prompt. Additionally, other works [31, 39, 43] incorporate a vision encoder to handle input images and [7] leverages a large multimodal model. Similarly to these, we treat the layout generation task as a code generation problem and utilize VLMs to process image and text inputs.

3. The value of sketches as a guidance method

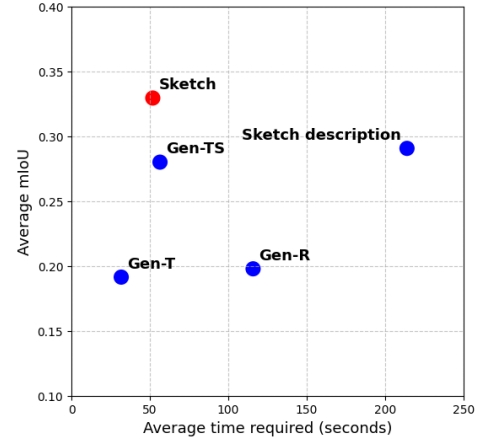


Figure 2. Time-performance trade-off between guidance methods on the PubLayNet dataset.

We start by assessing empirically the value of user-defined sketches as a guidance method for layout generation. We use a few-shot ($k=32$) prompted Gemini 1.5 Pro [35] and compare the sketch, encoded as an image, to three textual guidance methods from prior work [26]: generation conditioned on asset types (Gen-T), generation conditioned on asset types and sizes (Gen-TS), generation conditioned on spatial relationship between assets (Gen-R). We also compare the efficacy of the sketch to a detailed textual description of the sketch, generated by a captioning model. Details on few-shot prompt construction and the format for every guidance method are provided in the supplementary material.

We evaluate performance using the maximum Intersection over Union (*mIoU*), i.e. the largest possible IoU over all the possible matchings between generated and reference assets. More information about this metric can be found in Sec. 5.1. We use the same three datasets as in our experiments. To quantify the time efficiency of each guidance method, we measure the average time required to provide the input. For sketch-based guidance, we measure the time taken to collect each stroke, while for textual constraints, we estimate the time required to write the prompt assuming a typing speed of 200 characters per minute.

The time-performance trade-off of each guidance method measured on our largest dataset, PubLayNet [41], is shown in Fig. 2. The results clearly demonstrate the superiority of sketches for guiding layout generation, which maximizes performance while at the same time minimizing the time required to form the guidance signal. We report additional visualizations for the other datasets in the supplementary material.

4. Methodology

Inspired by previous work in the literature [23, 26], we formulate the sketch-to-layout problem as a code generation task. Layouts are encoded as protocol buffer strings [1], with attributes describing the position of assets and their properties. This code representation of the layouts enables a language modeling formulation of the problem. The flexibility of the protocol buffer format allows for straightforward conversion to SVG and therefore image rendering. Outputting a structured representation has several advantages over outputting a layout directly in pixel-space: (i) it can be verified that there are no hallucinations w.r.t. the input assets, (ii) the output can be easily interpreted and edited and (iii) it enables interoperability with existing creation tools e.g. document editors using a structured representation.

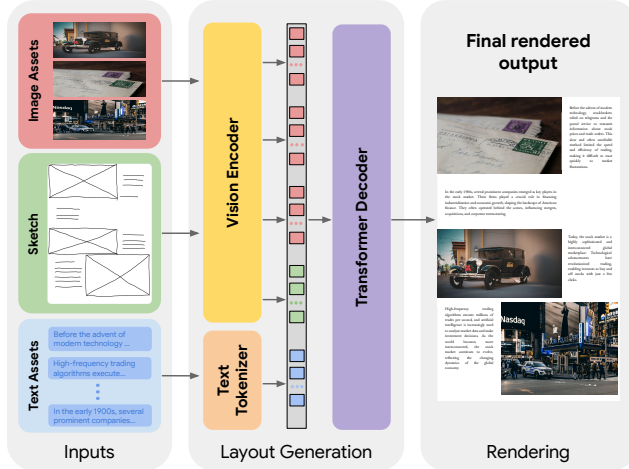


Figure 3. Our method: a sketch, alongside image and text assets are given to a VLM which generates the structured representation format of the layout, which can be rendered as an image.

Fine-tuning a VLM to perform well on the novel task of sketch-guided layout generation requires a large amount of human-drawn sketches paired with layouts. Such large-scale data collection is costly and time-consuming. In this section, we first discuss the open-source VLM we adopted to solve the layout generation task and then we introduce a scalable methodology to generate synthetic sketches for model training, while requiring a minimum amount of human data annotation.

4.1. Model Structure

To tackle the problem, we fine-tune PaLiGemma 3B [3], an open-source VLM trained to be versatile and effective to transfer. The language backbone of the architecture consists of Gemma [36], a decoder-only transformer pre-trained on code generation tasks. This makes PaLiGemma well suited for our layout generation task.

The model is multimodal, enabling us to provide both visual and textual inputs to guide the layout generation. An ink-based hand-drawn sketch, outlining the layout structure, is fed into the vision encoder alongside relevant image assets that should appear in the final layout; see also Fig. 3. Similarly to previous works on applications of VLMs to videos, the visual backbone of the model is applied independently on each input image, and the patch embeddings are concatenated. Therefore, the ViT [9] serves as a feature extractor, for both the sketch and image assets. The fact that our model processes image and text content allows the model to understand where to place the assets on canvas, generating a coherent narrative flow.

In addition, a textual prompt specifying the desired layout dimensions, asset names and the content of textual elements, is given as input to the VLM. An example of the prompt can be found in the supplementary material.

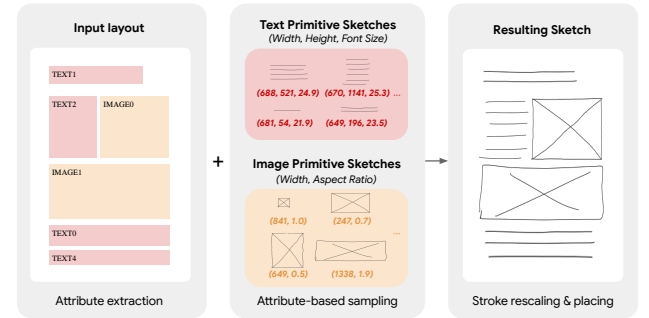


Figure 4. **Synthetic Sketch Generation Pipeline.** Every asset is matched with a stroke primitive based on its attributes and strokes are rescaled and combined to generate the synthetic sketch.

4.2. Synthetic Sketch Generation

Training the model on sketch-to-layout tasks requires paired data in the form of handwritten sketches and graphic layouts. To the best of our knowledge, there are no publicly available datasets with handwritten sketches resembling the layouts' structure. However there are document [30, 41] and slide [33] datasets, which are particular forms of layouts, without sketches. As direct human sketch annotation at the scale of these datasets is prohibitively costly, we introduce a scalable way to compose a relatively small number of human annotated sketches of layout elements into whole layout synthetic sketches. The methodology is in two steps.

Primitive collection. First, we collected a set of handwritten primitives for image and text assets. Inspired by wireframing, we defined primitives for image and text elements, using one or more horizontal lines to represent a block of text, and a crossed-out rectangle to represent an image. We sampled a set of text and image assets and asked 10 human annotators to draw ink-based sketch primitives on top of these assets, using tablet devices and a custom data

collection app. In total, we collect 237 primitives for training and 236 for validation. Since our data consists of image and text assets, we distinguish these two types of primitives. However, the exact same methodology could be extended to an arbitrary number of primitive types for more complex datasets.

Synthetic Sketch Composition. For a given layout, we compose a synthetic sketch by combining sketch primitives for every asset in the layout. More specifically, for each asset, we select a set of k candidate primitives that are the closest in terms of euclidean distance computed on the standardized width and aspect ratio for images and bounding box width, height and font size for texts. Then, from the k closest primitives, we select one at random. The process is conceptually illustrated in Fig. 4.

This methodology doesn’t require costly human sketching of full layouts and the annotation time scales linearly with the number of primitives, rather than the number of samples in the training set. It took 50 minutes to collect all the primitives necessary to construct our datasets. Meanwhile, it took on average 51.85 seconds to collect a full sketch for a PubLayNet sample for our test set, 36.54 seconds for DocLayNet and 13.56 seconds for SlideVQA. Assuming the same average time required for training and validation sets, collecting sketch data for our datasets would have required 2336, 292 and 63 human hours for PubLayNet, DocLayNet and SlideVQA respectively. More details on the implementation are in the supplementary material.

By using this novel procedure, we obtain a large dataset of layouts paired with sketches: 175k from PubLayNet, 33k from DocLayNet and 27k from SlideVQA. More dataset information and implementation details are provided in the appendix.

5. Experiments

5.1. Experiment Setup

Dataset Name	# Training Set	# Validation Set	# Human-Collected Test Set
PubLayNet	162 192	900	251
DocLayNet	28 780	900	268
SlideVQA	16 593	900	249

Table 1. Dataset Statistics. While train and validation splits contain synthetically created sketches, the test sets consist of human collected sketches.

Datasets. We conduct experiments on three publicly available datasets: PubLayNet [41], DocLayNet [30] and SlideVQA [33]. Dataset statistics are summarized in Table 1. We carry out training and hyper-parameter tuning on data paired with synthetic sketches, generated as described previously. Training details are provided in the supplementary material. To fairly assess model performance under

real-world conditions, we also collect real human-annotated sketches which we use as a test set.

We train separate models on each dataset and compare their performance to established baselines (detailed in the following section). While training a single model across all datasets could leverage cross-dataset knowledge and potentially improve performance, using separate models for each dataset ensures a fair comparison with baselines, which were exposed to individual datasets only.

Baselines. Since the sketch-to-layout task is a novel task and currently under-explored, there is no ‘ideal baseline’ designed to tackle exactly this problem. For this reason, we compare our method against a set of closely related baselines, in order to assess and put our model’s performance in perspective with alternative solutions.

We compare our method to LayoutPrompter [26], a recent method for conditioned layout generation. LayoutPrompter handles different types of constraints to generate layouts through few-shot prompting. Since *text-davinci-003*, the LLM used by LayoutPrompter, is now deprecated, we substitute it with a few-shot prompted Gemini 1.5 Pro [35], making this baseline even stronger.

LayoutPrompter use the following guidance methods: generation conditioned on asset types (Gen-T), generation conditioned on asset types and sizes (Gen-TS), generation conditioned on spatial relationship between assets (Gen-R). These constraints provide different levels of layout information. More details on these guidance methods and few-shot prompt examples are provided in the supplementary material.

In order to show the value of our synthetic data, we also compare our approach using a fine-tuned small model to a sketch-guided state-of-the-art VLM, Gemini 1.5 Pro with few-shot prompting.

An important difference between our method and LayoutPrompter is that our method is content-aware: it takes as input the sketch *and* the text and image assets. On the contrary, LayoutPrompter’s inputs consists of only layout constraints. In order to provide a fair comparison with the baselines, and analyse the effect of providing the content of the assets, we report results in both the content-agnostic and content-aware settings. For the no-content setting, we train our model without providing images and text assets. For the content-aware setup, we add asset content to the few-shot examples for both LayoutPrompter and few-shot Gemini baselines.

Metrics. To evaluate model and baseline performance, we use metrics widely adopted in the literature.

Intersection over Union (IoU) and *Maximum Intersection Over Union (mIoU)* [18]. Differently from IoU, where every generated asset is matched to the target asset sharing

the same name or identifier, mIoU corresponds to the maximum intersection over union over all the possible matchings between generated and target assets, where elements are paired only depending on their position.

IoU and mIoU are usually limited when evaluating unconstrained layout generation as they reward the model for generating a layout resembling the target, which might not be the only correct way to generate a visually appealing layout with the given assets. However, in our approach, the user explicitly guides the model towards the target layout using a sketch, specifying where the assets should be positioned. This guidance makes IoU an appropriate metric to measure model performance.

Overlap [22] measures the percentage of overlap between generated assets. *Alignment* [22] measures the graphical alignment for the layout. These metrics are commonly used in the literature. Note, however, that lower values for these metrics are not necessarily better in case of constrained layout generation. Alignment may not be always in agreement with the user intent as depicted by the sketch.

As we claim our approach is content-aware, it is necessary to introduce metrics measuring this awareness, rather than only focusing on the geometric structure of the layout.

Content Ordering Score. Inspired by [22], we introduce a new metric leveraging the Levenshtein Distance [20] to measure if the ground truth reading order and narrative flow are preserved in the generated document, taking values between 0 and 1. To compute the Levenshtein Distance, we take the center of each asset’s bounding box, and sort them first by Y-coordinate, then by X-coordinate. This aligns with the intuition of reading in left-to-right orientation languages: assets are sorted top-to-bottom and left-to-right. Then, for a set of asset names $\{a_k\}_{k=1}^n$, sorted as described above by their center coordinates, we map every asset a_k to a string character $c(a_k)$ and create a sequence $y = \text{concat}(c(a_1), \dots, c(a_n))$. The Content Ordering Score (COS) is computed as

$$COS = 1 - \frac{\text{lev}(\hat{y}, y)}{\max(|\hat{y}|, |y|)},$$

where $\hat{y}$ is the layout generated by the model, y is the ground truth layout and $\text{lev}(\cdot)$ is the edit distance between assets in two layouts.

5.2. Main Results

In what follows, we present the main results of this work. We compare our method against prior techniques, providing results highlighting the effectiveness of our proposed approach. We also provide some ablations studies which deepen our understanding of the method’s behaviour.

5.2.1. Content-Aware Solution

For the content-aware layout generation setting, we compare our approach to Gemini-based LayoutPrompter

(Gen-T, Gen-TS and Gen-R) and sketch-guided few-shot prompted Gemini. Results are presented in the Table 2.

Our model significantly surpasses the alternative approaches in terms of Maximum IoU, almost achieving a 50% improvement. This result, alongside qualitative results in the supplementary material, demonstrates our method’s ability to correctly place elements within the canvas, carefully following the user-provided sketch structure.

When evaluating our model on synthetic and human-produced sketches, we find comparable performance, demonstrating a minimal distribution shift between our synthetic sketches and full human-produced sketches. We report the results in Table 3. This result validates the use of synthetic sketches as proper training data for fine-tuning VLMs on the sketch-to-layout problem.

5.2.2. The Importance of Content-Awareness

To further assess the effectiveness of our content-aware approach, we create a comparative experiment using an approach not leveraging asset content. This allows to directly measure the benefits of incorporating content information. We fine-tune PaliGemma providing the sketch as the only visual input and an auxiliary textual prompt. Differently from before, image and text assets are not provided at this stage, and the textual prompt only contains information about the layout dimensions and asset types. Adopting the same hyper-parameters as before, we fine-tune three separate models, one per dataset. Results are reported in Table 4.

Despite surpassing baseline performance, the sketch-only method is unable to match the results achieved by the content-aware model. This suggests that including asset content information boosts performance, as expected. A visual example illustrating the typical improvements obtained through content-awareness can be found in the supplementary material.

5.3. Ablation Studies

To rigorously assess the efficacy of our proposed methodology, we conducted a series of ablation studies.

5.3.1. Partial Sketches

In our experiments so far we have assumed complete sketches i.e., the sketch covers all assets of the layout. To further assess the model’s performance, we introduced scenarios with partial sketches that cover only a subset of layout elements. This allows us to evaluate the model’s creative potential when faced with incomplete information.

The way a partial sketch is generated is the following: given a coverage rate p and the set of assets in a layout, each one of them is randomly included in the sketch with probability p . We experiments with coverage rates of 0%, 25%, 50%, 75% and 100%. An example of partial sketch for different coverage rates is reported in Figure 5.

Method	PubLayNet					DocLayNet					SlidesVQA				
	IoU $\uparrow$	COS $\uparrow$	mIoU $\uparrow$	Align. $\downarrow$	Overlap $\downarrow$	IoU $\uparrow$	COS $\uparrow$	mIoU $\uparrow$	Align. $\downarrow$	Overlap $\downarrow$	IoU $\uparrow$	COS $\uparrow$	mIoU $\uparrow$	Align. $\downarrow$	Overlap $\downarrow$
LayoutPrompter(Gen-T) w/ content	0.13	0.52	0.21	0.04	0.18	0.13	0.55	0.19	1.75	0.03	0.39	0.68	0.43	1.77	2.47
LayoutPrompter(Gen-TS) w/ content	0.14	0.27	0.29	0.29	0.08	0.14	0.38	0.22	2.52	0.09	0.40	0.57	0.44	6.92	2.49
LayoutPrompter(Gen-R) w/ content	0.11	0.49	0.22	0.25	0.15	0.12	0.56	0.19	0.79	0.11	0.35	0.68	0.39	0.98	2.35
Sketch-guided Gemini w/ content	0.15	0.33	0.32	0.31	0.08	0.15	0.42	0.25	0.93	0.05	0.40	0.63	0.46	1.85	2.43
FT-PaLiGemma w/ content (Ours)	0.62	0.69	0.76	0.34	0.03	0.46	0.68	0.59	2.92	0.03	0.66	0.79	0.75	6.54	2.42

Table 2. Comparison between Content-Aware FT-PaLiGemma and content-aware baselines. $\uparrow$ indicates larger values are better, $\downarrow$ indicates smaller values are better. Alignment values are multiplied by 1000, while Overlap results are multiplied by 10.

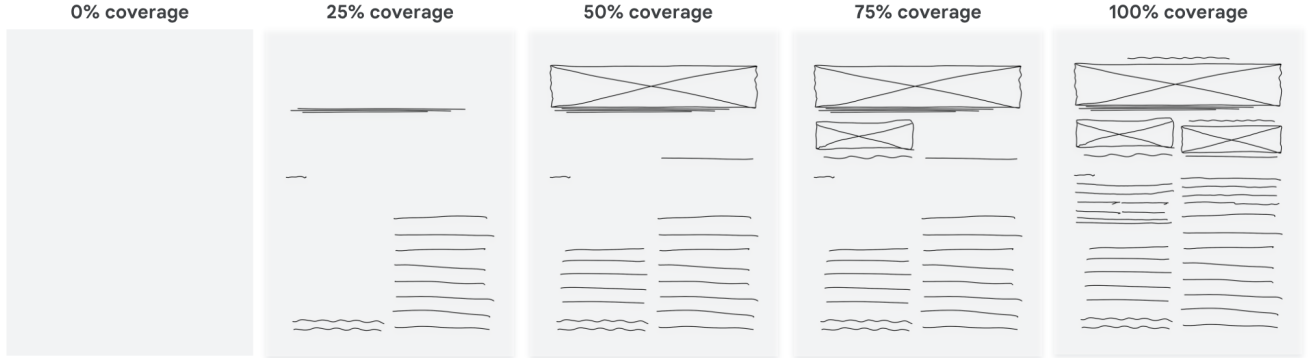


Figure 5. Different coverage rates.

Method	DocLayNet		PubLayNet		SlideVQA	
	IoU $\uparrow$	COS $\uparrow$	IoU $\uparrow$	COS $\uparrow$	IoU $\uparrow$	COS $\uparrow$
Synthetic	0.47	0.67	0.68	0.74	0.64	0.75
Human	0.46	0.66	0.62	0.70	0.66	0.79

Table 3. Performance comparison of our model on the test set on human sketches vs. synthetic sketches.

The ablation results are provided in Figure 6. We notice a clear trend: increasing sketch coverage correlates with improved IoU, confirming again the sketch’s role as a valuable constraint. Similarly, the Content Ordering Score (COS) increases with coverage, as expected due to the sketch’s guidance in asset positioning. Notably, both plots show that the content-aware model consistently outperforms its sketch-only counterpart across all coverage levels, confirming our previous findings.

5.3.2. Content ablations

We examine our setting more carefully to identify whether the model exploits shortcuts that can be present in the data. Such shortcuts can help the model place assets without looking at the asset content. We identified and tested a number of settings in which the model can exploit a spurious correlation.

- *Gibberish text*: text asset contents are replaced by random strings containing Latin letters, digits, and whitespaces, aimed at investigating if the model exploit the text sequence length to place text assets.

- *Random images with dimensions*: we replace image pixels with Gaussian noise, keeping the image dimensions in the prompt. This setting potentially eliminates the impact of image dimensions in asset placement.
- *Random images without dimensions*: image pixels are replaced with Gaussian noise, and image dimensions are removed from the prompt. In this setting, no information about the image is given to the model.
- *Full Content*: the original setting where the model has both text and image contents.

Results are shown on Figure 7. On PubLayNet and DocLayNet, the gibberish text setting performs worse than the original setting in terms of both IoU and COS, suggesting that model does not use text length as a shortcut. On random image settings, we see that removing images does not necessarily lead to a drop in performance. This limitation of our approach can be partially explained by the fact that in our datasets the majority of examples have one image, and only a small part has two or three images, and placing one image given a sketch is a trivial task. Moreover, the PaLiGemma [3] model has not been pretrained on multiple *uncorrelated* images, and achieving understanding of multiple images through a short fine-tuning on a narrow-domain dataset is a difficult task.

6. Conclusion

In this work, we present an end-to-end sketch-guided approach for layout generation leveraging VLMs. We motivate the choice of sketches as a guidance method, inspired

Method	PubLayNet			DocLayNet			SlidesVQA		
	mIoU $\uparrow$	Align. $\downarrow$	Overlap $\downarrow$	mIoU $\uparrow$	Align. $\downarrow$	Overlap $\downarrow$	mIoU $\uparrow$	Align. $\downarrow$	Overlap $\downarrow$
LayoutPrompter(Gen-T) w/o content	0.22	0.10	0.12	0.18	0.48	0.08	0.39	0.74	2.58
LayoutPrompter(Gen-TS) w/o content	0.33	0.37	0.08	0.24	1.95	0.08	0.43	4.28	2.40
LayoutPrompter(Gen-R) w/o content	0.23	0.36	0.19	0.18	0.79	0.12	0.36	1.45	2.49
Sketch-guided Gemini w/o content	0.33	0.46	0.02	0.23	0.23	0.03	0.47	2.76	2.39
FT-PaliGemma w/o content (Ours)	0.67	0.34	0.03	0.60	2.75	0.03	0.71	6.73	2.44

Table 4. Comparison between Sketch-Only FT-PaliGemma and content-agnostic baselines. $\uparrow$ indicates larger values are better, $\downarrow$ indicates smaller values are better. Alignment values are multiplied by 1000, while Overlap results are multiplied by 10.

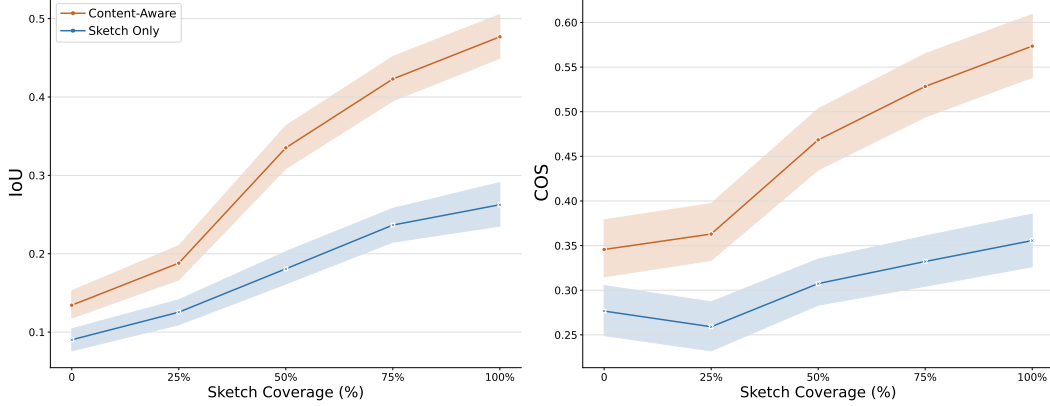


Figure 6. Content-aware metrics for different coverage rates of partial sketches, measured on DocLayNet. The blue and the orange line shows the content-agnostic (i.e., sketch-only) and content-aware comparison correspondingly.

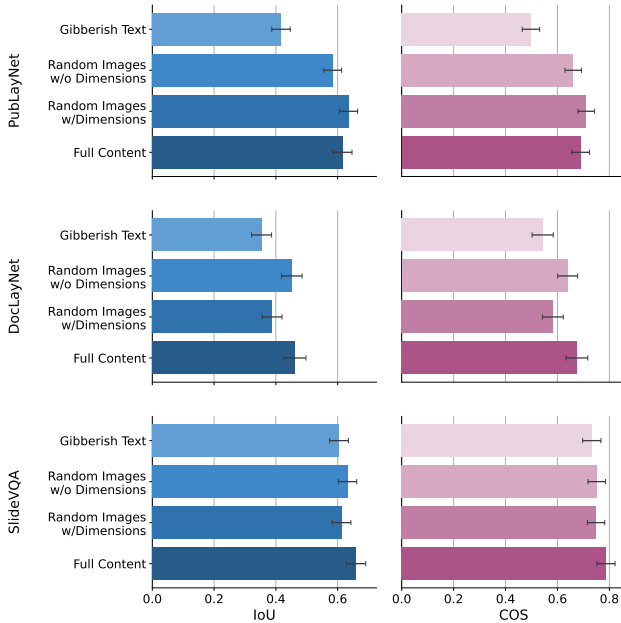


Figure 7. Intersection Over Union and Content Order Score on different settings ablating content.

with other guidance methods. To train our model, we introduce a novel technique to create synthetic sketches that requires only a few hours of human annotation work and can be scaled to cover large datasets. We release both the human annotated test set and the synthetic train set of sketches. Our fine-tuned model is content-aware and outperforms other constraint-based layout generation methods. Our approach can be generalized to different datasets and domains. More complex sketch primitives can also be added to further guide the model. We encourage future work to apply this methodology to generate sketches for a variety of domains and asset types and train larger, more powerful models to achieve production-level performance.

References

- [1] Protocol buffer documentation. <https://protobuf.dev/.4>
- [2] Daniel Baulé, Christiane Gresse Von Wangenheim, Aldo Von Wangenheim, Jean CR Hauck, and Edson C Vargas Júnior. Automatic code generation from sketches of mobile applications in end-user development using Deep Learning. *arXiv preprint arXiv:2103.05704*, 2021. 3
- [3] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda

by how UX designers work, through a comparative analysis

- Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. PaliGemma: A versatile 3B VLM for transfer, 2024. [2](#), [4](#), [7](#)
- [4] Bill Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan kaufmann, 2010. [2](#)
- [5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end Object Detection with Transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. [3](#)
- [6] Chin-Yi Cheng, Forrest Huang, Gang Li, and Yang Li. PLayer: Parametrically Conditioned Layout Generation using Latent Diffusion. In *ICML*, 2023. [2](#)
- [7] Yutao Cheng, Zhao Zhang, Maoke Yang, Hui Nie, Chunyuan Li, Xinglong Wu, and Jie Shao. Graphic Design with Large Multimodal Model. *arXiv preprint arXiv:2404.14368*, 2024. [3](#)
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019. [3](#)
- [9] 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. [3](#), [4](#)
- [10] Joao Silva Ferreira, André Restivo, and Hugo Sereno Ferreira. Automatically Generating Websites from Hand-drawn Mockups. In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2021. [3](#)
- [11] 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. [2](#)
- [12] Hsiao Yuan Hsu, Xiangteng He, Yuxin Peng, Hao Kong, and Qing Zhang. PosterLayout: A new Benchmark and Approach for Content-aware Visual-textual Presentation Layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6018–6026, 2023. [3](#)
- [13] 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 (CVPR)*, pages 10167–10176, 2023. [2](#)
- [14] 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. [3](#)
- [15] Vanita Jain, Piyush Agrawal, Subham Banga, Rishabh Kapoor, and Shashwat Gulyani. Sketch2Code: Transformation of Sketches to UI in Real-time using Deep Neural Network. *arXiv preprint arXiv:1910.08930*, 2019. [3](#)
- [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, 2023. [2](#)
- [17] 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. [2](#)
- [18] Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Constrained Graphic Layout Generation via Latent Optimization. In *Proceedings of the 29th ACM International Conference on Multimedia*. ACM, 2021. [2](#), [5](#)
- [19] 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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 491–506. Springer, 2020. [2](#)
- [20] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10: 707, 1966. [6](#)
- [21] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. LayoutGAN: Synthesizing Graphic Layouts with Vector-wireframe Adversarial Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7): 2388–2399, 2020. [2](#)
- [22] Jianan Li, Jimei Yang, Jianming Zhang, Chang Liu, Christina Wang, and Tingfa Xu. Attribute-conditioned Layout GAN for Automatic Graphic Design, 2020. [2](#), [6](#)
- [23] Ryan Li, Yanzhe Zhang, and Diyi Yang. Sketch2Code: Evaluating Vision-Language Models for Interactive Web Design Prototyping. *arXiv preprint arXiv:2410.16232*, 2024. [2](#), [4](#)
- [24] Xudong Liang and Tao Lin. Sketch2Wireframe: an automatic framework for transforming hand-drawn sketches to digital wireframes in UI design. *The Visual Computer*, pages 1–11, 2023. [3](#)
- [25] Jiawei Lin, Jiaqi Guo, Shizhao Sun, Weijiang Xu, Ting Liu, Jian-Guang Lou, and Dongmei Zhang. A Parse-Then-Place Approach for Generating Graphic Layouts from Textual Descriptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23622–23631, 2023. [2](#), [3](#)
- [26] Jiawei Lin, Jiaqi Guo, Shizhao Sun, Zijiang Yang, Jian-Guang Lou, and Dongmei Zhang. LayoutPrompter: Awaken the Design Ability of Large Language Models. *Advances in Neural Information Processing Systems*, 36, 2024. [2](#), [3](#), [4](#), [5](#)
- [27] Soumik Mohian and Christoph Csallner. Doodle2App: Native app code by freehand UI sketching. In *Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems*, pages 81–84, 2020. [3](#)

- [28] Brad Myers, Sun Young Park, Yoko Nakano, Greg Mueller, and Amy Ko. How Designers Design and Program Interactive Behaviors. In *2008 IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 177–184. IEEE, 2008. 2
- [29] Mark W Newman and James A Landay. Sitemaps, Storyboards, and Specifications: A Sketch of Web Site Design Practice. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 263–274, 2000. 2
- [30] 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*. ACM, 2022. 1, 2, 4, 5
- [31] Jaejung Seol, Seojun Kim, and Jaejun Yoo. PosterLlama: Bridging Design Ability of Language Model to Contents-Aware Layout Generation. *arXiv preprint arXiv:2404.00995*, 2024. 3
- [32] Mohammad Amin Shabani, Zhaowen Wang, Difan Liu, Nanxuan Zhao, Jimei Yang, and Yasutaka Furukawa. Visual Layout Composer: Image-Vector Dual Diffusion Model for Design Layout Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9222–9231, 2024. 3
- [33] Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. SlideVQA: A Dataset for Document Visual Question Answering on Multiple Images, 2023. 1, 2, 4, 5
- [34] 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. 2, 3
- [35] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sericinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdiah, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Matteo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurmurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshov, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayana Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton,

Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeynep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Cohen, Clemens Meyer, Katerina Tsihlias, Ada Ma, Juraj Gotweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Prolev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak

Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiujia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Girdano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khushen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xi-ang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirschall, Weiye Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C. Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhuyun Xiao, Rupert Kemp,

Huaixiu Steven Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Wojciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard, Achintya Singhal, Thang Luong, Boyu Wang, Sujevan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarakal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Koppurapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturk, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Vilella, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Rudnick, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnappalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Bal-

aji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Wooyeol Kim, Nandita Dukkkipati, Anthony Baryshnikov, Christos Kaplanis, XiangHai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecznikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Kiam Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Chojui Hsieh, Irene Cai, Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Bidsy, Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patrascu, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrin, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kepa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. Gemini 1.5: Unlocking Multimodal Understanding Across Millions of Tokens of Context, 2024. 3, 5

- [36] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha

Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikula, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. [4](#)

- [37] Ying Cao Xinru Zheng, Xiaotian Qiao and Rynson W.H. Lau. Content-aware Generative Modeling of Graphic Design Layouts. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2019)*, 38, 2019. [3](#)
- [38] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5481–5489, 2021. [2](#)
- [39] Tao Yang, Yingmin Luo, Zhongang Qi, Yang Wu, Ying Shan, and Chang Wen Chen. PosterLLaVa: Constructing a Unified Multi-modal Layout Generator with LLM. *arXiv preprint arXiv:2406.02884*, 2024. [2](#), [3](#)
- [40] Ning Yu, Chia-Chih Chen, Zeyuan Chen, Rui Meng, Gang Wu, Paul Josel, Juan Carlos Niebles, Caiming Xiong, and Ran Xu. LayoutDETR: detection transformer is a good multimodal layout designer. *arXiv preprint arXiv:2212.09877*, 2022. [3](#)
- [41] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. PubLayNet: Largest Dataset Ever for Document Layout Analysis, 2019. [1](#), [2](#), [3](#), [4](#), [5](#)
- [42] Min Zhou, Chenchen Xu, Ye Ma, Tiezheng Ge, Yuning Jiang, and Weiwei Xu. Composition-aware Graphic Layout GAN for Visual-textual Presentation Designs. *arXiv preprint arXiv:2205.00303*, 2022. [3](#)
- [43] Wanrong Zhu, Jennifer Healey, Ruiyi Zhang, William Yang Wang, and Tong Sun. Automatic Layout Planning for Visually-Rich Documents with Instruction-Following Models. *arXiv preprint arXiv:2404.15271*, 2024. [3](#)