

# Multimodal Markup Document Models for Graphic Design Completion

Kotaro Kikuchi<sup>1</sup> Naoto Inoue<sup>1</sup> Mayu Otani<sup>1</sup> Edgar Simo-Serra<sup>2</sup> Kota Yamaguchi<sup>1</sup>  
<sup>1</sup>CyberAgent <sup>2</sup>Waseda University  
kikuchi\_kotaro\_xa@cyberagent.co.jp

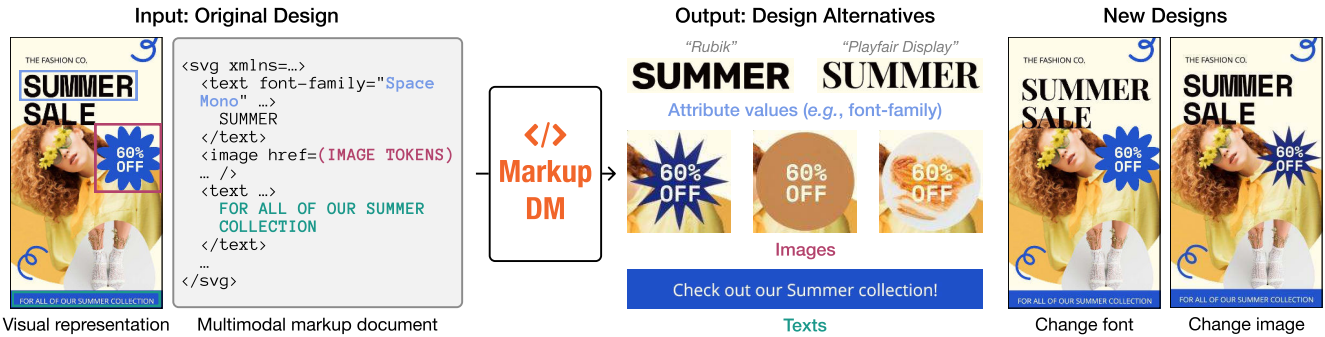


Figure 1. We present a multimodal markup document model (MarkupDM) for graphic design documents. Our model can generate alternative designs by inferring target spans, such as attribute values, images, and texts, from the surrounding context.

## Abstract

This paper presents multimodal markup document models (MarkupDM) that can generate both markup language and images within interleaved multimodal documents. Unlike existing vision-and-language multimodal models, our MarkupDM tackles unique challenges critical to graphic design tasks: generating partial images that contribute to the overall appearance, often involving transparency and varying sizes, and understanding the syntax and semantics of markup languages, which play a fundamental role as a representational format of graphic designs. To address these challenges, we design an image quantizer to tokenize images of diverse sizes with transparency and modify a code language model to process markup languages and incorporate image modalities. We provide in-depth evaluations of our approach on three graphic design completion tasks: generating missing attribute values, images, and texts in graphic design templates. Results corroborate the effectiveness of our MarkupDM for graphic design tasks. We also discuss the strengths and weaknesses in detail, providing insights for future research on multimodal document generation.<sup>1</sup>

## 1. Introduction

Graphic design is a visual medium that communicates information and ideas by effectively organizing text, images, and other elements. While graphic design appears everywhere in various applications, such as websites, advertisements, and printed materials, creating high-quality designs requires expertise and time. Several studies apply machine learning techniques to automate design tasks, such as layout generation [8–10, 19, 26, 27, 32], colorization [14, 22, 23], or typography stylization [29, 38]. Apart from the elaboration of individual tasks, there have been holistic modeling of graphic design to solve multiple tasks [11, 35]. While these approaches capture implicit and complex relationships among elements and their attributes, they suffer from limited training samples and have not yet achieved production-level performance. Recent studies show promising performance improvement for layout generation by utilizing the prior knowledge of Large Language Models (LLMs) [19, 26, 32]. The successful use of LLMs for a design task inspires our approach to using multimodal language models for graphic design.

In this paper, we investigate multimodal LLMs for holistic modeling of graphic design. We treat graphic design as an interleaved multimodal document consisting of markup language and images. This representation allows us to directly apply multimodal LLMs, which has shown promis-

<sup>1</sup>Project page: <https://cyberagentailab.github.io/MarkupDM/>

ing results in vision-and-language tasks [36], to the graphic design domain. We adopt the code LLM, which learned markup languages like HTML, as the base model to promote better knowledge transfer in structural expressions. One obstacle to practically applying multimodal LLMs is that while handling opaque and fixed-size images is common in the vision community, graphic design often involves images of different sizes with transparency, which makes it challenging to employ off-the-shelf image generators and quantizers. We address this problem by tailoring an image quantizer for graphic design.

We build our model, named the *markup document model* (*MarkupDM*), through two-stage training; first, we train an image quantizer that encodes transparent and different-sized images into discrete tokens, and then we extend a pre-trained code LLM to input and output image tokens. We train the MarkupDM on 166K graphic design templates based on the next token prediction. We transform the input sequence into the prefix-suffix-middle format so that the model can generate the missing middle part from the given prefix and suffix [1, 2].

We evaluate our MarkupDM on three graphic design completion tasks: generating missing attribute values, images, and texts in graphic design templates. Our MarkupDM can generate plausible designs consistent with the given context, allowing us to explore various design alternatives as shown in Fig. 1. We further discuss the strengths and weaknesses of our MarkupDM in detail and provide insights for future research on multimodal design generation.

In summary, our contributions are as follows:

- We propose a multimodal markup language model, a new class of multimodal LLM that considers graphic designs interleaved multimodal documents.
- To adapt LLMs into the graphic design application, we build an image quantizer that encodes images of different sizes with transparency into discrete tokens.
- We empirically show our MarkupDM can effectively solve graphic design completion tasks.

## 2. Related Work

### 2.1. Graphic Design Generation and Completion

There has been an ongoing research effort in computational support for graphic design, such as layout generation [8–10, 18, 19, 26, 27, 32, 34], colorization [14, 22, 23], typography stylization [29, 38], or general stylization [28]. Several previous works, like ours, infer missing parts or alternative solutions from the surrounding context. Completing a layout from a given partial layout is one of the common subtasks in layout generation [10]. Zhao *et al.* [38]

propose a model that predicts typographic styles in web design from visual and semantic contexts. Recently, Shao *et al.* [28] have presented a generative model for web page styling. Qiu *et al.* [23] propose a model based on the masked prediction on multi-palette color representation for recoloring graphic design documents.

Some studies aim to model the entire graphic design document. CanvasVAE [35] is a variational autoencoder that generates heterogeneous attributes such as type, position, size, and image content for each element in a graphic design document. FlexDM [11] adopts the masked prediction approach to efficiently capture relationships among elements and their attributes. Both models estimate the features of images and texts and then retrieve the most similar ones from the dataset. There is also active research into other approaches, such as generating stylized text that is placed on top of generated raster images [12, 13, 34], to generate high-quality overall designs.

A few works have started to apply large language models (LLMs) for graphic design tasks [18, 19, 26, 32]. Lin *et al.* [18] translate a text description of the desired layout into an intermediate text representation using LLM to guide the subsequent layout generation. LayoutNUWA [32] treat layout generation as a code generation task and utilize knowledge of LLMs to generate layout code. Concurrently, Cheng *et al.* [3] propose a multimodal LLM that generates attributes such as position and size for each element given as RGBA images.

Inspired by these studies, we propose a new approach for the holistic modeling of graphic design documents that leverages prior knowledge of multimodal LLM. In this context, our work can be seen as the first attempt to obtain image and text content by generation rather than by retrieval.

### 2.2. Multimodal Large Language Models

The recent success of LLMs has led to the development of multimodal LLMs that can recognize and generate images [36]. Several multimodal LLMs, such as GILL [15], Emu [30], and DreamLLM [5], are designed to connect LLMs with a off-the-shelf pre-trained image encoder, such as CLIP [24], and a decoder such as Stable Diffusion [25]. These pre-trained image encoders and decoders are challenging for graphic design completion tasks as they do not support transparent images. Training these encoders and decoders requires a large-scale dataset of image-text pairs. However, collecting such a dataset is challenging as images in graphic design documents are difficult to describe accurately with text.

Another approach of multimodal LLMs is representing images as discrete tokens [1, 4, 33]. This approach encodes images into a sequence of tokens via a pre-trained image quantizer, such as VQGAN [6]. Although the publicly available pre-trained quantizers often do not support images

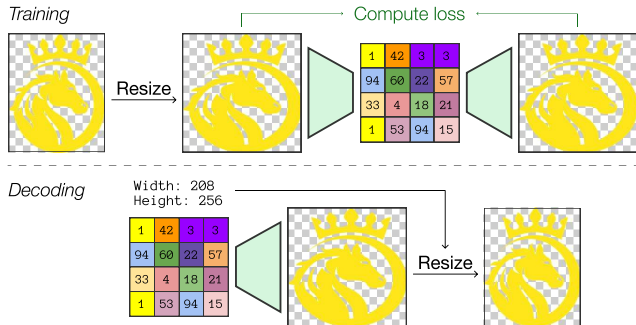


Figure 2. Our image quantizer is trained by reconstructing images resized to a fixed size. When decoding, the image size is given in addition to the image tokens.

with transparency, they only require images for training, not image-text pairs. We adopt this approach and tailor an image quantizer for graphic design.

### 3. Method

We first train an image quantizer to encode images into discrete tokens. We then build the MarkupDM by fine-tuning a pre-trained code LLM with interleaved multimodal documents to incorporate the image modality. We illustrate the overview of our method in Fig. 2 and Fig. 3.

#### 3.1. Image Quantizer

We train an image autoencoder that encodes images of different-sized images with transparency into discrete token maps with  $1/f$  resolution and decodes them back to the original images. In preliminary experiments, we found that varying the token size according to the image size makes it challenging to train the markup language model in the later stage. Instead, we take a simple but effective approach of resizing the input image to a fixed square size. We follow the previous studies [6, 25] and take the same network architecture and training objectives for our autoencoder, with the only difference related to the alpha channel. We set the number of input/output channels to four and consider L1 reconstruction loss for all channels. When calculating the loss based on RGB-based external models, *e.g.*, the perceptual loss [37], we convert generated RGBA images to RGB images by alpha compositing on a white background. We initialize our model with the weights of a pre-trained RGB image quantizer. For the alpha channel weights, we use the mean values of the corresponding RGB weights.

#### 3.2. Document Representation

Once we train the image quantizer, we apply the quantizer and convert graphic design templates into unified sequence representations. Our data representation is based on

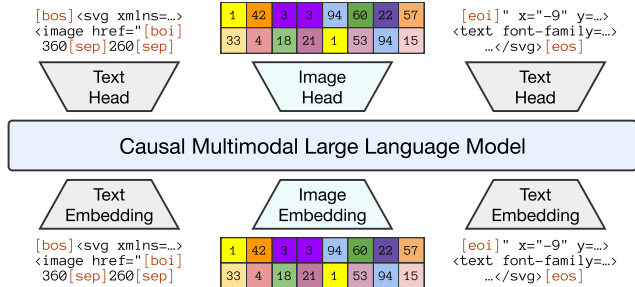


Figure 3. Our model is based on causal multimodal LLM, with separate embedding layers and prediction heads dedicated to images and text tokens.

the SVG format<sup>2</sup> with the difference that we embed discrete tokens obtained by quantizing the image instead of the path to the image file. We show an example of the markup document representation in the following:

```
Multimodal markup document

[ bos ]
<svg xmlns="http://www.w3.org/2000/svg" viewBo
x="0 0 4 19 298" width="419" height="298">
  <image href="[ boi ] 360 [ sep ] 260 [ sep ] [ img:1 ] [ im
g:42 ] [ img:3 ] [ img:94 ] ... [ eoi ] " x="-9" y="-9"
width="436" height="315" />
  <text font-family="Montserrat" font-size="30
" font-weight="bold" fill="rgba(255, 255, 25
5, 1)" x="32" y="81">FAMILY</text>
  ...
</svg>
[ eos ]
```

The image content, *i.e.*, the value of href attribute in the <image> tag, starts with the special token [boi] and ends with [eoi]. The inside of these is separated by the special token [sep], and each represents the width, height, and image tokens obtained by our image quantizer. This image representation is similar to the previous work on a multimodal LLM for simplified HTML documents [1], but differs in that the image size is also described as text and included in the target of generation.

#### 3.3. Multimodal Markup Language Model

We build MarkupDM based on the recent code LLMs, which are specifically tuned for handling coding tasks. We make two extensions to the base code LLM to incorporate the image representation described in Sec. 3.2. First, we extend the vocabulary of the base LLM to include the additional special tokens, such as [boi]. Second, we add new modules dedicated to the image tokens, such as [img:1], the embedding module, and the prediction head. In the embedding module, we first embed the image tokens via the frozen lookup table in our image decoder. We then concate-

<sup>2</sup><https://www.w3.org/TR/SVG11/>

nate them with the positional encodings [31] and project them to the same dimension as the text embeddings. The prediction head for image tokens is similar to text tokens, but uses a different set of parameters and vocabulary, *i.e.*, the codebook size in image quantization.

We train our model based on the next token prediction in our sequences to which we randomly apply the fill-in-the-middle transformation [1, 2], allowing the model to predict the missing middle part from the prefix and suffix parts. Our model has to know the next token’s modality for inference due to the different prediction heads. We determine the next modality on a rule based on the tokens generated so far.

## 4. Experiments

We first evaluate our image quantizer on the image reconstruction task and then our multimodal markup language models on several graphic design completion tasks.

### 4.1. Image Quantization

#### 4.1.1 Setup

We use an internal dataset on graphic design templates, similar to the Crello dataset [35]. The design template consists of an ordered set of elements, each associated with an element category, geometric attributes, and design attributes. The template also has global attributes such as canvas size. We use 800,000 RGBA images of non-textual elements in the design templates for training and 133,267 images from the other templates for evaluation.

We use the RGB image quantizer from Latent Diffusion Model [25] trained on the OpenImages dataset [16], which is mainly composed of photographs, as a baseline. Specifically, we use the quantizer with the scaling factor  $f = 16$  and the codebook size  $Z = 16,384$  based on the balance between the reconstruction quality and the resulting token length. We finetune the quantizer for 100,000 steps on our dataset using the techniques explained in Sec. 3.1 to adapt it to RGBA images. For further analysis, we finetune the quantizer solely using RGB images without special techniques, which we denote by *Ours-RGB*. We apply an off-the-shelf background removal tool to convert RGB images to RGBA images for comparison. We use Rembg [7] with the IS-Net model [21] for this purpose.

We evaluate the quantizers using the mean squared error (MSE) for RGB and alpha channels and the reconstruction Fréchet Inception Distance (rFID) for RGB images, which computes the distance between the feature distributions of the original and reconstructed images. We convert the RGBA images generated by our quantizer to RGB images by alpha compositing on a white background for RGB-based metrics.

Table 1. Quantitative comparison of image reconstruction using each quantizer. The lower the better for all metrics. The dagger † indicates the score when the alpha value for all pixels is set to 1.0.

	MSE		rFID
	RGB ( $\times 10^3$ )	Alpha ( $\times 10^{-1}$ )	RGB
Baseline [25]	2.42	3.75 <sup>†</sup>	6.34
Ours-RGB	<b>1.50</b>	3.75 <sup>†</sup>	<b>1.65</b>
Ours	<u>1.86</u>	<b>0.03</b>	<u>4.96</u>

Table 2. Quantitative comparison of infilling performance. The scores indicate accuracy, and the higher, the better. The *Font* represents the font family, and the *F-Size* represents the font size. FlexDM is formulated differently from ours, and the scores are not directly comparable and are only shown for reference.

	X	Y	Width	Height	Font	F-Size
FlexDM [11]	0.480	0.278	0.507	0.639	0.785	0.870
MarkupDM						
SC1-1B	0.541	0.379	0.827	0.895	0.769	0.673
SC1-3B	<u>0.587</u>	<u>0.444</u>	<u>0.865</u>	<u>0.936</u>	<u>0.807</u>	<b>0.728</b>
SC1-7B	<b>0.621</b>	<b>0.472</b>	<b>0.873</b>	<b>0.947</b>	<b>0.813</b>	<u>0.707</u>
SC2-3B	0.522	0.355	0.764	0.801	0.771	0.681
SC2-7B	0.548	0.380	0.773	0.800	0.780	0.705

#### 4.1.2 Image Reconstruction

We show the quantitative comparison of image reconstruction using each quantizer in Tab. 1. Both of our quantizers outperform the baseline in terms of RGB-based metrics thanks to fine-tuning images from the same domain. We show qualitative results in Fig. 4. We can see that RGB-based reconstruction with general background removal does not work well, as it removes foreground objects either excessively or insufficiently. In contrast, our quantizer successfully reconstructs RGBA images thanks to the alpha information embedded in the discrete tokens.

## 4.2. Graphic Design Completion

### 4.2.1 Setup

We use the same dataset as the image reconstruction task and convert 165,991 graphic design templates to SVG format to train the models. In the conversion, we represent text elements by `<text>` tags, and other elements by `<image>` tags. We do not specify attributes if they have the default values. Also, since SVG cannot render multi-line text in a single element, we split a text element into multiple elements when a new line appears.

We train our MarkupDM with the fill-in-the-middle (FIM) objective [1, 2] to predict the middle part of the sequence from the prefix and suffix parts. Considering



Figure 4. Image reconstruction results.

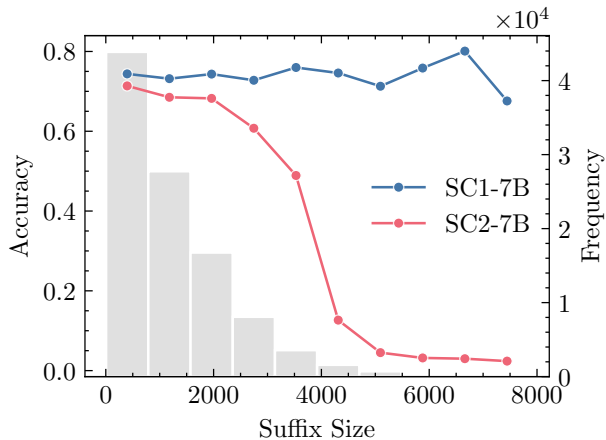


Figure 5. Mean accuracy of the infilled attribute values versus suffix length. The gray histogram shows the frequency distribution of suffix lengths.

the span we want to fill-in as the middle part, our model

can infer the span from the preceding and following context. We validate the effectiveness of our method with three tasks: *attribute value completion* represented like `<text_x="[MASK]"_...>` where [MASK] represents the target span to be filled, *image completion* represented like `<image_href="[MASK]"_.../>`, and *text completion* represented like `<text_...>[MASK]</text>`. We select six attribute types, `x`, `y`, `width`, `height`, `font-family`, and `font-size` for the attribute value completion task. Note that we do not train MarkupDM with task-specific supervision, such as task-specific FIM patterns, and tasks are post-hoc for evaluation purposes.

We evaluate MarkupDM using 13,188 to 24,872 target spans obtained from 2,000 unseen templates, selecting the corresponding spans for each task. We select StarCoder [17] (SC1) and StarCoder2 [20] (SC2) as the base LLMs for our MarkupDM for their strong performance on code comprehension and generation tasks. We are unaware of literature having the same experimental setting as ours, but for reference, we report the performance of the most



Figure 6. Image completion results. Each triplet shows the input, the predicted completion, and the original design from left to right or top to bottom. The [MASK] or [M] indicates the missing part to be completed.

similar method, FlexDM [11]. We train the FlexDM model on our dataset using random masking patterns and adjust it as comparable as possible. We also compare MarkupDM with the base LLMs in different model sizes. For attribute value completion, we report the mean accuracy for each attribute type. We parse attributes from the text generated by MarkupDM and evaluate the accuracy using the quantized representation used in FlexDM.

#### 4.2.2 Attribute Value Completion

We show the quantitative results on the attribute accuracy in Tab. 2. Note that the scores between FlexDM and MarkupDM are not fully comparable due to the different formulations and available contexts, *i.e.*, MarkupDM can predict element size using the image size, while FlexDM can not. MarkupDM performs reasonably well compared to FlexDM, showing that they are successfully trained to fill the graphic design templates. Among the MarkupDM, SC1-7B, the largest base LLM, yields the best performance



Figure 7. Text completion results shown similarly to Fig. 6. The green arrows point to the missing text, and the green boxes indicate the zoomed-in areas.

in most attributes, as expected. Using SC2, the successor of SC1, shows worse performance. SC2 adopts the sliding window attention (SWA) to reduce computational cost. We suspect that SWA may not be suitable for FIM style infilling, as the model can not know the valid contexts if the length of the suffix tokens is longer than the window size. Figure 5 shows the mean accuracy for each bin of the suffix length. The performance of SC2 indeed degrades significantly when the suffix length is longer than the window size (4,096 tokens).

### 4.2.3 Image and Text Completion

We perform image and text completion tasks using the best MarkupDM, SC1-7B. We show the qualitative results in Fig. 6 for successful image completion, Fig. 7 for successful text completion, and Fig. 8 for failure cases for both tasks.

We observe cases where the model generates plausible images by copying other images in the template, using repetition patterns and symmetry as hints, e.g., the top two examples in Fig. 6. Our model also succeeds in generating

typical decorations such as buttons and underlays, e.g., the two examples in the lower left. Our model is also good at generating background images that harmonize foreground objects, e.g., the bottom right example.

For text completion, our model can generate text that connects with the preceding or following lines in a grammatically correct way in many cases, e.g., the top left example in Fig. 7. Besides multiple lines, there are cases where the model generates text that matches the surrounding texts, e.g., the right example. The bottom left example does not have a strong textual context. Still, our MarkupDM successfully generates text with a typical role, using the position of the target element and the visual decoration as hints.

In the failure cases, MarkupDM has difficulty generating images of main objects due to the lack of context and poor image generation ability, e.g., the top left example in Fig. 8. It also fails to generate images that require delicate visual harmonization with other elements, e.g., the middle left example. For text completion, we can see failures due to errors in image understanding, e.g., “business school back-



Figure 8. Failure cases shown similarly to Figs. 6 and 7.

pack” is generated for the earbuds in the right example. It is not yet possible to generate text by adjusting its length so that it does not conflict with other visual decorations, e.g., the bottom left example.

### 5. Limitations and Discussion

We introduce MarkupDM, a new class of multimodal LLMs that can generate markup languages with transparent and different-sized images. We investigated MarkupDM’s performance in detail using three completion tasks, but it remains unclear whether our model can predict all aspects of an element, including its type, attributes, and content. We also have not verified whether our model can generate the

rest of the template when the first part is given. The primary limitation of MarkupDM is its poor ability to generate main images. Increasing the number of training images using other multimodal datasets may help improve the image generation quality. Alternatively, we could consider leaving the main image generation to an external strong image generator or adapting a strong RGB-based multimodal LLM to RGBA images. As extensions to MarkupDM, we are interested in settings that provide additional context via text prompts [12, 13] and incorporate reference designs via retrieval augmentation [8]. Finally, designing interfaces for multimodal LLMs, such as our MarkupDM, to support creative design workflows is another important direction that remains to be explored.



## References

- [1] Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. CM3: A Causal Masked Multimodal Model of the Internet. *arXiv preprint arXiv:2201.07520*, 2022. 2, 3, 4
- [2] Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient Training of Language Models to Fill in the Middle. *arXiv preprint arXiv:2207.14255*, 2022. 2, 4, 11
- [3] Yutao Cheng, Zhao Zhang, Maoge Yang, Hui Nie, Chunyuan Li, Xinglong Wu, and Jie Shao. Graphic design with large multimodal model. *arXiv preprint arXiv:2404.14368*, 2024. 2
- [4] Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*, 2024. 2
- [5] Runpei Dong, Chunrui Han, Yang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, Xiangwen Kong, Xiangyu Zhang, Kaisheng Ma, and Li Yi. DreamLLM: Synergistic multimodal comprehension and creation. In *ICLR*, 2024. 2
- [6] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming Transformers for High-Resolution Image Synthesis. In *CVPR*, 2021. 2, 3
- [7] Daniel Gatis. Rembg. <https://github.com/danielgatis/rembg>, 2020. (accessed 2024-08-27). 4, 5, 12
- [8] Daichi Horita, Naoto Inoue, Kotaro Kikuchi, Kota Yamaguchi, and Kiyoharu Aizawa. Retrieval-Augmented Layout Transformer for Content-Aware Layout Generation. In *CVPR*, 2024. 1, 2, 8
- [9] Mude Hui, Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, Yuwang Wang, and Yan Lu. Unifying layout generation with a decoupled diffusion model. In *CVPR*, 2023. 1, 2
- [10] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. LayoutDM: Discrete Diffusion Model for Controllable Layout Generation. In *CVPR*, 2023. 1, 2
- [11] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Towards flexible multi-modal document models. In *CVPR*, 2023. 1, 2, 4, 6
- [12] Naoto Inoue, Kento Masui, Wataru Shimoda, and Kota Yamaguchi. OpenCOLE: Towards Reproducible Automatic Graphic Design Generation. In *CVPRW*, 2024. 2, 8
- [13] Peidong Jia, Chenxuan Li, Yuhui Yuan, Zeyu Liu, Yichao Shen, Bohan Chen, Xingru Chen, Yinglin Zheng, Dong Chen, Ji Li, Xiaodong Xie, Shanghang Zhang, and Baining Guo. Cole: A hierarchical generation framework for multi-layered and editable graphic design. *arXiv preprint arXiv:2311.16974*, 2024. 2, 8
- [14] Kotaro Kikuchi, Naoto Inoue, Mayu Otani, Edgar Simo-Serra, and Kota Yamaguchi. Generative colorization of structured mobile web pages. In *WACV*, 2023. 1, 2
- [15] Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. Generating Images with Multimodal Language Models. In *NeurIPS*, 2023. 2
- [16] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Mallocci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4. *IJCV*, 128(7), 2020. 4
- [17] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Arnel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. StarCoder: May the source be with you! *TMLR*, 2023. 5
- [18] 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 *ICCV*, 2023. 2
- [19] 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 *NeurIPS*, 2023. 1, 2
- [20] Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo, Evgenii Zheltonozhskii, Nii Osae Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Arnel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian McAuley, Han Hu, Torsten Scholak, Sebastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, Mostofa Patwary, Nima Tajbakhsh, Yacine Jernite, Carlos Muñoz Ferrandis, Lingming Zhang, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. StarCoder 2 and The Stack v2: The Next Generation. *arXiv preprint arXiv:2402.19173*, 2024. 5
- [21] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly Accurate Dichotomous Im-

- age Segmentation. In *ECCV*, 2022. 4
- [22] Qianru Qiu, Xueting Wang, and Mayu Otani. Multimodal color recommendation in vector graphic documents. In *ACM MM*, 2023. 1, 2
- [23] Qianru Qiu, Xueting Wang, Mayu Otani, and Yuki Iwazaki. Color recommendation for vector graphic documents based on multi-palette representation. In *WACV*, 2023. 1, 2
- [24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021. 2
- [25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3, 4, 5, 12
- [26] Jaejung Seol, Seojun Kim, and Jaejun Yoo. PosterLlama: Bridging design ability of language model to contents-aware layout generation. In *ECCV*, 2024. 1, 2
- [27] 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 *CVPR*, 2024. 1, 2
- [28] Zirui Shao, Feiyu Gao, Hangdi Xing, Zepeng Zhu, Zhi Yu, Jiajun Bu, Qi Zheng, and Cong Yao. Webrpg: Automatic web rendering parameters generation for visual presentation. In *ECCV*, 2024. 2
- [29] Wataru Shimoda, Daichi Haraguchi, Seiichi Uchida, and Kota Yamaguchi. Towards diverse and consistent typography generation. In *WACV*, 2024. 1, 2
- [30] Quan Sun, Qiyang Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative Pretraining in Multimodality. In *ICLR*, 2024. 2
- [31] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *NeurIPS*, 2020. 4
- [32] Zecheng Tang, Chenfei Wu, Juntao Li, and Nan Duan. LayoutNUWA: Revealing the hidden layout expertise of large language models. In *ICLR*, 2024. 1, 2
- [33] Chameleon Team. Chameleon: Mixed-Modal Early-Fusion Foundation Models. *arXiv preprint arXiv:2405.09818*, 2024. 2
- [34] Haohan Weng, Danqing Huang, Yu Qiao, Zheng Hu, Chinyew Lin, Tong Zhang, and CL Chen. Design: A pipeline for controllable design template generation. In *CVPR*, 2024. 2
- [35] Kota Yamaguchi. CanvasVAE: Learning to generate vector graphic documents. In *ICCV*, 2021. 1, 2, 4
- [36] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A Survey on Multimodal Large Language Models. *arXiv preprint arXiv:2306.13549*, 2024. 2
- [37] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, 2018. 3
- [38] Nanxuan Zhao, Ying Cao, and Rynson W.H. Lau. Modeling fonts in context: Font prediction on web designs. *Computer Graphics Forum*, 37, 2018. 1, 2

## A. Implementation Details

We build our RGBA quantizer by finetuning a pre-trained RGB quantizer, as described in Secs. 3.1 and 4.1.1. We train our quantizer in 100,000 steps using mixed precision training using bfloat16 a batch size of 8 with a single NVIDIA L4 GPU. We use the Adam optimizer with a learning rate of  $1 \times 10^{-5}$ . We resize the input images to  $256 \times 256$  pixels. The training takes approximately 2 days to complete.

We build our MarkupDM by extending a pre-trained code LLM, as described in Secs. 3.3 and 4.2.1. We train our model in 100,000 steps with a single A100 80GB GPU with several techniques for efficient training, including mixed precision training using bfloat16, gradient checkpointing, and the Flash Attention 2. We use the Adam optimizer with a learning rate of  $5 \times 10^{-5}$  and a constant schedule. The fill-in-the-middle (FIM) transformation is applied with a probability of 0.9 during training. Specifically, we use the context-level FIM with the token-level span selection in the prefix-suffix-middle format [2]. The training takes approximately 1 day for SC1-1B version, 2 days for SC1-3B version, and 4 days for SC1-7B version to complete.

## B. Details on Our Experiments and Additional Results

We convert structured graphic design documents to SVG format, as described in Secs. 3.2 and 4.2.1. We show in Tab. 3 the element tags and their corresponding attributes used in the experiments.

Table 3. Element tags and their corresponding attributes used in our experiments. Please refer to the SVG specification for details.

Element tag	Attribute
svg	xmlns, viewBox, width, height
image	href, x, y, width, height, transform, opacity
text	x, y, fill, font-family, font-size, font-weight, font-style, text-anchor, letter-spacing, transform, opacity

For graphic design completion, we use the top- $p$  sampling with  $p = 0.9$  for generation. We set the maximum number of new tokens to 10 for attribute value completion, 50 for text completion, and 278 for image completion (256 for the image tokens, 2 for the special tokens, and 20 for the image width and height).

We provide additional image reconstruction results in Fig. 9 and design completion results in Figs. 10 and 11 by MarkupDM (SC1-7B). We set the temperature to 2.0 to generate more diverse outputs for the font type completion.



Figure 9. Additional image reconstruction results. Our quantizer ( $f = 16$ ) can encode RGBA images so that they can be reconstructed with high fidelity. Reconstructing human faces is still challenging for our quantizer, but it may be alleviated by using finer scaling factors or additional losses for faces.

**PERSONALITY. PERSONALITY.**  
**PERSONALITY. PERSONALITY.**

YOUR HOUSE IS THE REFLECTION OF YOUR PERSONALITY.  
 We are open for mid-season sale. You can find the latest collections

Link in bio

Get 40% off for your first design  
 Extra 20% off for all item

You can find the latest collections  
 Enjoy your visit!

Home LOREM  
**DECOR**  
 BESPOKE, HANDMADE & SUSTAIN

**DECOR DECOR**  
**DECOR DECOR**

**BESPOKE, HANDMADE & SUSTAIN**

From marvelous masterpieces  
 Your house is a dream home  
 Let's decorate your home with love!

DR. LEO JONES  
**PROMO COUPON**  
 ONLY **\$149**  
 PER VISIT

**PROMO COUPON**  
 Toothbrush  
 Vitamins  
 NEW TREATMENT

ONLY **\$149** PER VISIT  
 ONLY **\$149** PER VISIT  
 ONLY **\$149** PER VISIT  
 ONLY **\$149** PER VISIT

SHOP NAME

CELEBRATING BASTILLE DAY

ONE DAY SALE  
 UP TO **50% OFF**

SHOP NOW WWW.SHOPXARE.COM


SHOP NOW SHOP NOW SHOP NOW SHOP NOW

**50% OFF** 50% 70% OFF 30% OFF

Figure 10. Additional qualitative results for design completion. The left most designs are the original designs, and the right figures show the results of attribute value (font type), text, and image completion. The green boxes indicate the original design to be altered.

Name your brand

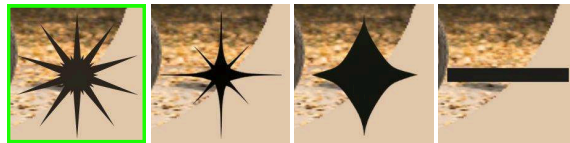
# BICYCLES RENTAL



We accept bicycle rental at affordable prices.

**BOOK NOW** [www.reallygreatsite.com](http://www.reallygreatsite.com)

**BICYCLES** BICYCLES  
BICYCLES BICYCLES



**BOOK NOW** [www](http://www) **Order Now** [www](http://www)  
**+123-456-7890** [www](http://www) **More info check our site** [www](http://www)

## ONLINE DANCE CLASSES

EVERY WEEKEND 7AM TO 8AM


**UP TO 50% OFF**

**DANCE CLASSES** **DANCE LESSON** **DANCE FOR KIDS** **DANCE DANCE**



Create your unique **EASTER BASKET**


**ONLINE!**



**15% off**

SAME DAY DELIVERY

**SHOP NOW**



**ONLINE!**

**ONLINE!**

**ONLINE!**

**ONLINE!**

**EASTER BASKET**  
EASTER BASKET  
EASTER BASKET  
EASTER BASKET

**SAME DAY DELIVERY**

Get a 30% discount for purchasing

Choose your size and toppings

Special offer only this week

Figure 11. Additional qualitative results for design completion (continued).