MATCHA : Enhancing Visual Language Pretraining with Math Reasoning and Chart Derendering

Fangyu Liu[♠♣*] **Francesco Piccinno**[♣] **Syrine Krichene**[♣] **Chenxi Pang**[♣] **Kenton Lee**[♣]
Mandar Joshi[♣] **Yasemin Altun**[♣] **Nigel Collier**[♠] **Julian Martin Eisenschlos**[♣]
[♣]Google DeepMind [♠]University of Cambridge

Abstract

Visual language data such as plots, charts, and infographics are ubiquitous in the human world. However, state-of-the-art vision-language models do not perform well on these data. We propose MATCHA (**Math** reasoning and **Cha**rt derendering pretraining) to enhance visual language models' capabilities in jointly modeling charts/plots and language data. Specifically we propose several pretraining tasks that cover plot deconstruction and numerical reasoning which are the key capabilities in visual language modeling. We perform the MATCHA pretraining starting from Pix2Struct, a recently proposed image-to-text visual language model. On standard benchmarks such as PlotQA and ChartQA, the MATCHA model outperforms state-of-the-art methods by as much as nearly 20%. We also examine how well the MATCHA pretraining transfers to domains such as screenshots, textbook diagrams, and document figures and observe overall improvement, verifying the usefulness of MATCHA pretraining on broader visual language tasks.[1][2]

1 Introduction

Visual language is the system that uses tightly integrated textual and visual elements to convey meaning (Horn, 1998). It is ubiquitous in the human world with typical examples being charts, plots and diagrams existing in places such as textbooks, scientific papers web pages and many more. Visual language is also highly complex – besides texts, its structural units can include line, shape, color, orientation, scale, angle, space, etc. One needs to recognize patterns from these structural units, and perform spatial grouping and/or alignment to extract information for reasoning.

Whilst being prevalent and important, there is little research on visual language understanding from the machine learning community. Vision-language models pretrained on natural images or image-text pairs crawled from the web perform badly on visual language tasks such as ChartQA (Masry et al., 2022) and PlotQA (Methani et al., 2020) due to the high complexity of jointly modeling language and symbols (more evidence in experiments). Pix2Struct (Lee et al., 2023) is a recently proposed pretraining strategy for visually-situated language that significantly outperforms standard vision-language models, and also a wide range of OCR-based pipeline approaches. Pix2Struct designs a novel masked webpage screenshot parsing task and also a variable-resolution input representation for pretraining an image-to-text encoder-decoder Transformer (Vaswani et al., 2017). In this work, we use Pix2Struct as the base model and further pretrain it with chart derendering and math reasoning tasks.

We argue that visual language understanding needs two key ingredients: (1) layout understanding (including number extraction and their organizations) and (2) mathematical reasoning. (1) is required to discover the underlying patterns of the image and organize the elements in the image in a logical form. (2) is needed to operate on the elements extracted from (1) and derive meaningful information demanded by a task or query. Based on these observations, we propose two complementary pretraining tasks for enhancing visual language understanding: **chart derendering** and **math reasoning**. In chart derendering, given a plot/chart, the image-to-text model is required to generate its underlying data table or the code used to render it. The second task is math reasoning pretraining. We pick two numerical reasoning dataset MATH (Saxton et al., 2019) and DROP (Dua et al., 2019), render the input into images and the image-to-text model needs to decode the answers.

*Work done during Google internship.

[1]Code and models: github.com/google-research/google-research/tree/master/deplot

[2]For questions paper please contact fl399@cam.ac.uk and eisenjulian@google.com.

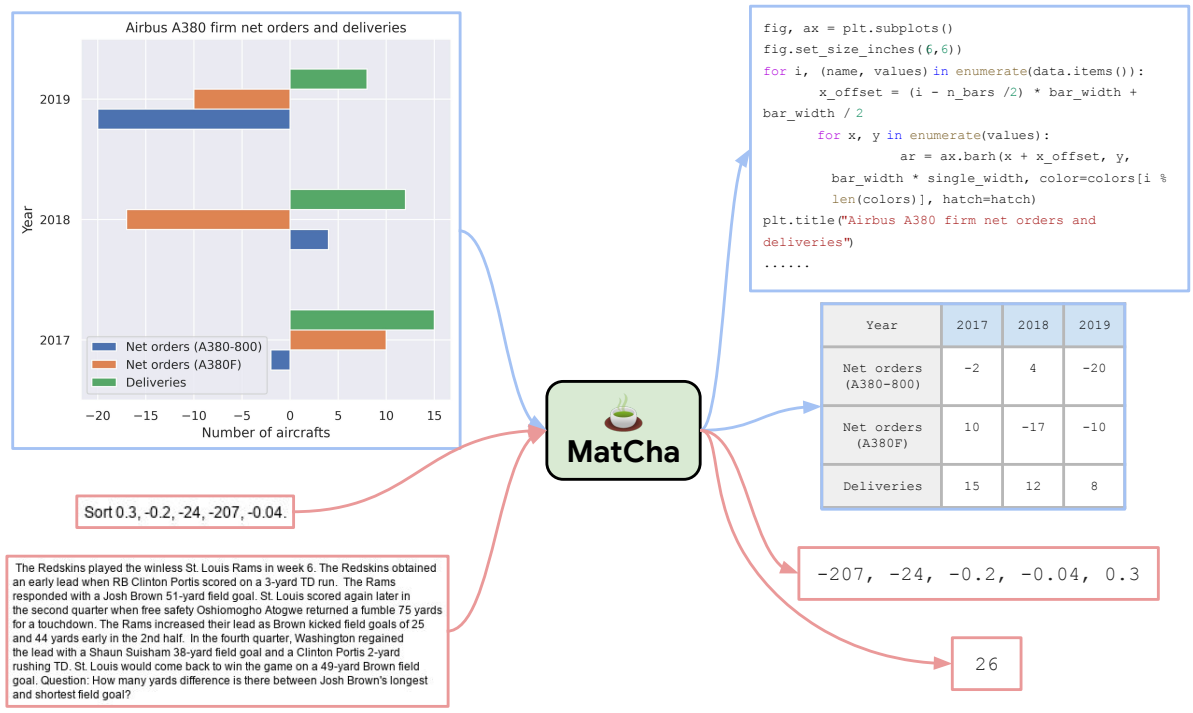


Figure 1: MATCHA defines two types of pretraining tasks: (1) chart derendering (light blue boxes) and (2) mathematical reasoning (light red boxes). In chart derendering, given a chart, the model needs to decode its underlying rendering code or data table. In math reasoning, given a math question rendered as an image, the model needs to decode its answer. Chart derendering teaches the models layout understanding (including number extraction and their organizations) and math reasoning teaches the models numerical reasoning capabilities.

We use a suite of visual language tasks to test the effectiveness of our method. Most importantly, we test on ChartQA and PlotQA which are QA datasets about plots and charts. On both datasets, MATCHA surpasses even the SOTA model assuming access to charts' underlying data tables and can beat the prior SOTA without gold data table by as much as 20%. We also test MATCHA on chart-to-text summarization tasks and observe clear improvements over Pix2Struct and achieves SOTA on Chart-to-Text (Kantharaj et al., 2022) Pew split. Last but not least, to examine if the MATCHA pretraining generalizes to datasets beyond the standard plots and charts domain, we also test MATCHA on four additional domains where Pix2Struct was evaluated on: documents, illustrations, user interfaces, and natural images (including datasets, such as textbook QA, Widget Captioning, etc.). We demonstrate consistent improvement on most additional datasets compared with the base model Pix2Struct.

To summarize, our contributions are: (1) proposing a set of effective pretraining tasks for visual language learning (2) demonstrating consistent improvements across all evaluated tasks and SOTA results on ChartQA, PlotQA, and Chart-to-Text summarization (Statista set) without accessing the gold data tables; (3) verify that MATCHA pretraining transfers to visual language benchmarks beyond the chart & plot domains and achieve SOTA across a wide range of datasets beyond the chart domain such as textbook VQA and Widget Captioning; (4) comprehensive ablation and analyses to understand the effect of each pretraining component and its impact to downstream performance.

2 Related Work

Vision-language research and a lack of attention on visual language. Research on vision-and-language has predominately been focusing on natural images. Visually-grounded reasoning datasets such as NLVR2 (Suhr et al., 2019) and MaRVL (Liu et al., 2021) are mostly in the natural image domain. Synthesized datasets such as SHAPES (Andreas et al., 2016), NLVR (Suhr et al., 2017), and CLEVR (Johnson et al., 2017) can be seen as in the visual language domain. However, their visual language systems are significantly simpler than those in the real world such as plots and charts. As a result, information extraction from these synthesized datasets is straightforward. Besides, the

queries in the synthesized datasets are relatively naive and do not require complex reasoning (e.g., questions can usually be on spatial relations or counting objects). Consequently, current vision-language models can handle the above mentioned synthesized visual reasoning datasets quite well. However, they do not perform well on real-world visual language datasets where both the information extraction and reasoning becomes much more complex (we will show this in §4).

OCR-based & end-to-end methods for visually-situated language.[3] LayoutLM (Xu et al., 2020; Huang et al., 2022) leverages a patch-OCR alignment loss to inject an external OCR systems' knowledge into the Transformer model. PresSTU (Kil et al., 2022) and PaLI (Chen et al., 2023) also design OCR-aware pretraining objectives where the model needs to predict texts obtained from off-the-shelf OCR systems. ChartBERT (Akhtar et al., 2023) relies on OCR text and positions to train a transformer encoder. While OCR systems can be helpful for accurately extracting texts, running them is not cheap. Also, OCR systems do not cover visual language systems that do not explicitly use text. As examples, plots and charts do not always have numbers written explicitly. In our concurrent work DEPLOT (Liu et al., 2023), we explore combining a chart-to-text translation module (without OCR) with large language models.

Donut (Kim et al., 2022), Dessurt (Davis et al., 2023), and Pix2Struct (Lee et al., 2023) are end-to-end pretrained models for visual language where Donut and Dessurt focus on document understanding and Pix2Struct aim to provide a generic pretrained checkpoint for all visual language tasks. MATCHA's architecture is identical to Pix2Struct – we continually pretrain a Pix2Struct checkpoint with new objectives.

Learning to reason by designing novel pretraining tasks. MATCHA is related to the literature of designing better pretraining objectives to help the language models (LMs) to reason better since the skill is hard to require through naive language modeling objectives only (e.g, masked language modeling and autoregressive language modeling on raw text). Geva et al. (2020); Eisenschlos et al. (2020) generate additional pretraining data focused on (numerical) reasoning through human-written templates. Pi et al. (2022) synthesize data and programs, and then use program executors to simulate answers. LMs are pretrained to predict the answers given data and programs. Wu et al. (2022) explore a wide range of synthetic pretraining tasks and found that even just injecting knowledge as simple as induction and deduction rules could teach LMs to reason. We teach an image-to-text model to reason through mapping charts to data and code, and also directly learning textual math reasoning datasets.

3 Method

We argue that layout understanding and basic math operation capabilities are the key elements for performing visual language understanding/reasoning. We inject such capabilities to the model by proposing two pretraining tasks: **chart derendering** (§3.1) and **math reasoning** (§3.2) which we introduce in detail in the following sections.

3.1 Chart Derendering

Plots and charts are usually generated by an underlying data table and a piece of code. Code decides the overall layout of the figure (e.g., type, direction, color/shape scheme of the chart) and the underlying data table decides the actual numbers and the groupings of them. Both the data and code are sent to a compiler/rendering engine to create the final image. To understand a chart one needs to discover the visual patterns in the image, effectively parse and group them to extract the key information. Reversing the plot rendering process demands all such capabilities and can thus serve as a perfect pretraining task.

In practice, it is challenging to simultaneously obtain charts, their underlying data tables, and their rendering code. To collect sufficient pretraining data, we independently accumulate (chart, code) and (chart, table) pairs. For (chart, code), we crawl all GitHub IPython notebooks with appropriate licenses and extract blocks with figures. A figure and the code block right before it are saved as a (chart, code) pair.[4] For (chart, table) pairs, we explored

[3] There is a nuanced difference between *visual* language and *visually-situated* language. Most models discussed here are specifically designed for images with significant amount of texts (e.g., documents) and thus they are models primarily for visually-situated language. Visual language data significantly overlaps with visually-situated language data. However, visual language also covers the scenarios where no/few texts are explicitly used but visual objects/patterns are most responsible for presenting information (e.g., certain types of plots).

[4] Note that the code snippet can be noisy since earlier blocks could also be relevant for generating the figure and also the snippet may contain bits of code that is irrelevant to generating the figure. Also note that the data table is fre-

two sources. First is to manually write code for converting web-crawled Wikipedia tables from Herzig et al. (2020) to charts. We randomly combine several plotting options. The key random variables include: using either `matplotlib` or `seaborn` as the plotting package; using either bar, line, or pie chart; styles and colors of the charts; whether to show numbers explicitly on the graph; font and size of the texts. Besides our own synthetic data, we also add chart-table pairs generated by Methani et al. (2020) (from PlotQA) to diversify the pretraining corpus. The second source is web-crawled chart-table pairs. Websites such as Statista provides both. We directly use the chart-table pairs crawled by Masry et al. (2022) (from ChartQA), containing around 20k pairs in total from four websites: Statista, Pew, Our World in Data, and OECD.[5]

Note that to avoid leaking test information for the PlotQA and ChartQA tasks which use the same chart data as pretraining, we only use the chart-table pairs in the training sets for pretraining and test tables/charts are strictly excluded. In ablation study (§5.1), we will show that chart-table from both sources are useful and having a diverse set of chart-table pairs is always better. However, using only our synthetic data brings very significant improvement already, suggesting that the concept of chart derendering can be easily transferred to charts of other domains (including real-world charts).

3.2 Math Reasoning

Reasoning over visual language requires (1) effective recognition and grouping of the visual elements and also (2) applying mathematical operations (such as sorting, min/max, averaging, etc.) on top of them. Plot derendering addresses (1) but (2) is still lacking in the current pretraining framework. As a result, we propose to explicitly inject numerical reasoning knowledge to the image-to-text model by learning from textual math datasets.

We use two existing textual math reasoning datasets, MATH (Saxton et al., 2019) and DROP (Dua et al., 2019) for pretraining. MATH is synthetically created, containing two million training examples per module (type) of questions (see Appx. §A for a comprehensive listing of modules included in MATCHA pretraining). DROP is a reading-comprehension-style QA dataset where the input is a paragraph context and a question. DROP has 96k question and answer pairs over 6.7K paragraphs.[6] To solve questions in DROP, the model needs to read the paragraph, extract relevant numbers and perform numerical computation to predict the answer. We found both datasets to be complementarily helpful. MATH contains large amounts of questions and is categorized which helps us identify math operations needed to explicitly inject to the model. DROP's reading-comprehension format resembles the typical QA format where models need to simultaneously perform information extraction and reasoning. In practice, we render inputs of both datasets into images (concatenating the context and question for DROP). The image-to-text model is trained to decode the answer given the redered image. Examples of MATH and DROP can be found in Figure 1 (in light red).

Besides the two newly proposed pretraining strategies, to prevent catastrophic forgetting, we also keep applying the screenshot parsing pretraining from Pix2Struct (Lee et al., 2023). Specifically, given screenshot of a website (where parts of the website is masked), the image-to-text transformer needs to predict the underlying simplified HTML code that could render the original unmasked website screenshot. The final pretraining task is a mixture of all aforementioned tasks. We discuss the mixture weights in §4.1.

4 Experiment

We detail our experimental setup in §4.1, introduce the main results in §4.2, and results on additional Pix2Struct tasks in §4.3.

4.1 Experimental Setups

Pretraining datasets/tasks. Overall, we create a mixture of pretraining task that has 40% of math reasoning, 40% of chart derendering, and 20% screenshot parsing. The weight for specific task/dataset is listed in Table 1. For chart derendering, we have four sources of data: (1) chart-table pairs synthesized by ourselves, (2) from ChartQA, (3) synthesized in PlotQA, and (4) chart-to-code data. We initially assigned equal weight to the four tasks however noticed training instability since chart-to-code is very hard (the pretraining data is noisy). We thus lower chart-to-code to 4% and increase all chart-to-table tasks to 12%. For math reasoning, we assign equal weights to MATH and

quently missing and usually not hardcoded in the notebook. As a result, we collect (chart, table) pairs separately.

[5] See Appx. §A for links.

[6] Note that for all datasets used for pretraining, we always use only the training set if there exists a split.

Component	Task/Dataset	Rate	Size
Math reasoning	MATH dataset	20%	2M
	DROP	20%	96K
Chart derendering	Chart-to-code (GitHub; **ours**)	4%	23M
	Chart-to-table (synthetic; **ours**)	12%	270K
	Chart-to-table (ChartQA)	12%	22K
	Chart-to-table (PlotQA)	12%	224K
Pix2Struct	Screenshot parsing	20%	80M

Table 1: Mixture rates for all tasks in pretraining and the absolute size of each dataset. The mixture rate is used to sample each example within the batch.

Task	Dataset	# Tables	# Pairs
Chart Question Answering	ChartQA (Human)	4.8K	9.6K
	ChartQA (Machine)	17.1K	23.1K
	PlotQA (v1)	224K	8M
	PlotQA (v2)	224K	29M
Chart Summarization	Chart-to-Text (Pew)	9K	9K
	Chart-to-Text (Statista)	35K	35K

Table 2: Statistics of the finetuning datasets.

DROP (both are 20%).

For pretraining dataset ablation studies, see §5.1.

Evaluation datasets. We evaluate MATCHA on multimodal English QA and generation tasks including ChartQA (Masry et al., 2022), PlotQA (Methani et al., 2020),[7] and Chart-to-Text summarization (Kantharaj et al., 2022). Both ChartQA and PlotQA are chart domain QA datasets where the input is an image of a chart and a query and the target is an answer string. ChartQA has two subsets: (1) augmented and (2) human where the augmented set is machine generated and thus more extractive and the human set is human written and requires more complex reasoning. PlotQA also has two sets v1 and v2. Similarly, v1 focuses more on extractive questions and v2 requires more numerical reasoning. However, both v1 and v2 are machine generated. Chart-to-Text has two sets as well. They are "Pew" and "Statista" where the names describe the source of the image examples. For Pew, the gold summaries are automatically extracted from areas around the image. For Statista, the summaries are human written. The sizes of each dataset are described in Table 2.

Beyond chart domain datasets, we additionally evaluate on other datasets used in Pix2Struct (Lee et al., 2023). We follow the exact same setups and protocols of Pix2Struct by rerunning Pix2Struct experiments but replacing the initial checkpoint with MATCHA. See Lee et al. (2023) for more experimental details.

Metrics. For ChartQA and PlotQA, following previous works (Masry et al., 2022; Methani et al., 2020; Lee et al., 2023), we use relaxed correctness (exact match but tolerating 5% of numerical error). For Chart-to-Text, we use BLEU4. For all Pix2Struct experiments, we use identical metrics introduced in Lee et al. (2023).

Training and inference details. We save checkpoint every 200 steps and keep the checkpoint that produces the highest validation score. Following Lee et al. (2023), we finetune models on the ChartQA aug. and human sets together (i.e., one checkpoint for two sets) and use the checkpoint selected on human val set as the final checkpoint for testing. For PlotQA and Chart-to-Text, we train standalone models for v1, v2, Pew, and Statista sets. For pretraining, we use a batch size of 512 and max sequence length of 192. We pretrain for 100k steps and the final MATCHA checkpoint is selected at the 90k step (where the average exact match validation score is the highest). For downstream tasks finetuning, we use a batch size of 256 and max sequence length of 128. For ChartQA and Chart-to-Text we finetune for 10k steps and for PlotQA we finetune for 20k steps (since it is significantly larger). Setups for Pix2Struct tasks are the same as the original paper. As for the PaLI baselines, we use the larger 17B variant and finetune for 5k steps and save checkpoints every 1000 steps. All MATCHA and Pix2Struct models are pretrained/finetuned with 64 GCP-TPUv3 while PaLI models are finetuned with 128 GCP-TPUv4.

Note that since MATCHA is an image-to-text model (without a textual input branch), whenever it is required to input text to the model, the text is rendered as an image. As an example, for QA tasks, we prepend the question as a header above the chart and input the image with question header as a whole to the model.

4.2 Main Results

We summarize the main results in Table 3 where we compare MATCHA with a wide range of baselines and SOTA models[8] across three chart/plot-domain benchmarks ChartQA, PlotQA, and Chart-to-Text Summarization. On ChartQA, MATCHA

[7] There exists othear chart domain QA datasets such as DVQA (Kafle et al., 2018) and FigureQA (Kahou et al., 2017). However, they are both synthetic and SOTA models have already reached $> 95\%$ accuracy. We thus focus on more challenging datasets.

[8] For brief introduction of baselines used, please see Appx. §B.

Model	Gold Table?	ChartQA			PlotQA			Chart-to-Text			avg. (all)
		aug.	human	avg.	v1	v2	avg.	Pew	Statista	avg.	
T5	yes	-	-	59.8	93.2	85.6	89.4	-	37.0	-	-
VL-T5	yes	-	-	59.1	**96.4**	84.7	90.6	-	-	-	-
VisionTaPas	yes	-	-	61.8	80.2	58.3	69.3	-	-	-	-
CRCT	no	-	-	-	76.9	34.4	55.7	-	-	-	-
VL-T5-OCR	no	-	-	41.6	75.9	56.0	66.0	-	-	-	-
T5-OCR	no	-	-	41.0	72.6	56.2	64.4	10.5	35.3	22.9	42.8
VisionTaPas-OCR	no	-	-	45.5	65.3	42.5	53.9	-	-	-	-
PaLI-17B (res. 224)	no	11.2	15.2	13.2	56.9	13.1	35.0	10.0	40.2	25.1	24.4
PaLI-17B (res. 588)	no	64.9	30.4	47.6	64.5	15.2	39.8	11.2	**41.4**	**26.3**	37.9
Pix2Struct	no	81.6	30.5	56.0	73.2	71.9	72.5	10.3	38.0	24.2	50.9
MATCHA	no	**90.2**	**38.2**	**64.2**	92.3	**90.7**	**91.5**	**12.2**	39.4	25.8	**60.5**

Table 3: Main experimental results on two chart QA benchmarks ChartQA & PlotQA and a chart summarization benchmark Chart-to-Text. Detailed introduction of the baselines can be found in Appx. §B.

beats the previous SOTA (without access to the underlying gold data table) Pix2Struct by 8.2%. Even if we consider models that do assume the existence of gold data tables, they generally underperform MATCHA by 3-5%. The best performing baseline VisionTaPas has a specialized module for modeling tables but still lags behind MATCHA by 2.4%. On PlotQA, MATCHA is again the best performing model overall. On the v1 set, VL-T5 with access to underlying data table performs better than MATCHA by $\approx 4\%$ which is intuitive since PlotQA is a synthetic dataset thus containing relative simple queries and the v1 is the extractive set where queries are even more straightforward. On v2 where questions are related to numerical reasoning, MATCHA outperforms all models including the models with access to underlying gold tables. On Chart-to-Text summarization, MATCHA improves upon Pix2Struct on both Pew and Staista and is the new SOTA on Pew. However, MATCHA underperforms PaLI-17B (res. 588) on Statista.

Overall, MATCHA is clearly the best-performing model with SOTA or competitive performance on every setup and all tasks. All baselines without access to gold tables lag behind significantly – MATCHA outperforms the strongest baseline without gold table access Pix2Struct by $\approx 10\%$ if we average the performance scores across all datasets.

Among the baselines, we would like to highlight PaLI which is the SOTA for a series of multimodal text-image tasks such as VQA and captioning on natural images and is of a much larger size (i.e., 17B parameters vs. 300M in MATCHA). PaLI fails significantly on ChartQA and PlotQA since the challenge in the visual language is distinctly different from that in the natural image domain. Increasing input resolution substantially helps the model's performance (likely due to the better text reading with higher resolution) but this also increases the sequence length (thus also memory and compute) quadratically. PaLI performs reasonably well in Chart-to-Text. We believe this is because the Chart-to-Text task (evaluated by BLEU4) might be more sensitive to textual fluency but less sensitive to factuality as compared with the other two QA tasks. It is expected that PaLI trained with a language modeling objective on natural text will have more advantage under this evaluation setup.

4.3 Results on Pix2Struct Tasks

Besides chart/plot domain datasets, we would also like to examine if MATCHA transfers to other visual language datasets such as documents, user interfaces, and natural images. We rerun all Pix2Struct finetuning experiments with a MATCHA checkpoint and the results are shown in Table 4. On average across all tasks, MATCHA outperforms Pix2Struct by 2.3%. Besides ChartQA, the improvement is also observed in AI2D (QA on textbook diagrams), Widget Captioning (recognizing and captioning widgets in screenshots), DocVQA (QA on scanned documents), etc. Even if we exlucde ChartQA, MATCHA can outperform Pix2Struct by 1.6% on average, suggesting that knowledge learned through MATCHA pretraining can be transferred to visual language domains out side of plots/charts.

Tasks→	ChartQA	AI2D	OCR-VQA	RefExp	Widget-Cap	Screen-2Words	Text-Caps	Doc-VQA	Info-VQA	avg.	avg. (excl. ChartQA)
Pix2Struct	56.0	40.9	**69.4**	92.2	133.1	**107.0**	88.0	72.1	**38.2**	77.4	80.1
MATCHA	**64.2**	**42.6**	68.9	**94.2**	**137.7**	106.2	**92.4**	**74.2**	37.2	**79.7**	**81.7**

Table 4: MATCHA vs. Pix2Sturct on Pix2Sturct tasks.

5 Analyses and Discussions

In this section, we first conduct pretraining ablations in §5.1 to understand the usefulness of each pretraining component, then in §5.2 we conduct fine-grained analysis and error analysis to probe MATCHA' strengths and weaknesses.

5.1 Ablation Study

Setup↓	aug.	human	avg.
MATCHA (full; 50k steps)	88.6	**37.4**	**63.0**
Component-level ablations			
- no math reasoning	88.2	33.0	60.6
- no chart derendering	83.7	34.4	59.1
- no Pix2Struct screenshot parsing	87.8	34.9	61.4
Single-task ablations			
- no MATH dataset	88.2	36.7	62.5
- no DROP dataset	88.2	34.3	61.3
- no real-world chart-table pairs	87.4	34.5	61.0
- no chart-to-code	**89.1**	34.6	61.9

Table 5: MATCHA pretraining ablations on ChartQA.

We conduct two types of ablations. First, we remove a whole type of pretraining datasets. For example, 'no math reasoning' means removing the whole math reasoning component and drops the MATH and DROP datasets. The weights of other datasets in the mixture are proportionally increased. Second, we remove an individual dataset within a component. For example, 'no MATH dataset' means removing just MATH dataset but keep other datasets in the math reasoning component untouched. In this scenario, we increase the weight of other math datasets (in this case just DROP) proportionally to maintain the overall weight of the component in the mixture. To reduce compute used, we train one full MATCHA model and all its ablated models with 50k steps (the original full MATCHA is trained for 100k steps). As a result the MATCHA model performance in Table 5 is slightly lower than the 100k model (63.0 vs. 64.2). The pretrained models are then finetuned and evaluated on ChartQA only. The full ablation study table is shown in Table 5 where the first half is component-level ablations and the second half is individual dataset ablation.

The impact of each pretraining component. On the component-level, we found that removing any major component (math reasoning, chart derendering, and screenshot parsing) would cause a performance drop. The most important component is chart derendering, the removal of which causes a decrease of $\approx 4\%$ averaging across the two sets. Removing math reasoning decreases the avg. score by 2.4% and removing the continual pretraining of screenshot parsing causes a drop of 1.6%. We notice that math reasoning is more important to the human set while chart derendering is more important on the augmented set. The findings are likely due to the fact that the human set contains more numerical reasoning questions while the augmented set contains more extractive questions. We also conducted ablations of specific datasets/tasks which we discuss in paragraphs below.

MATH vs. DROP dataset for learning to reasoning. We have used two datasets, i.e. MATH and DROP, for injecting numerical reasoning capabilities to MATCHA. According to Table 5, we observe that DROP seems to be more important (the removal of which causes a performance drop of 1.7% vs. a drop of 0.5% from the removal of MATH). We conjecture that it is because the reading-comprehension-QA format of DROP is more similar to the downstream task of QA on visual language, where information extraction and reasoning needs to be jointly performed.

Synthetic vs. real-world corpus as pretraining chart-table pairs. We perform another ablation to justify the choice of chart derendering pretraining corpus. Real-world chart-table pairs can increase the diversity and coverage of chart derendering pretraining however we need to explicitly scrape such data from the web. We are interested in understanding to what extent our manually syn-

thesized charts and plots with existing libraries can improve model's performance. The row 'no real-world chart-table pairs' shows results of only using synthesized chart-table data by us (i.e., no ChartQA and PlotQA chart-table data). The overall performance drops by 2%. Interestingly, for the augmented set, the performance only drops 1.2% but almost 3% is dropped on the human set. This indicates that extractive questions can usually be solved with synthetic pretraining but the more diverse real-world data (also usually having more sophisticated layout) can benefit reasoning learning more.

The impact of chart-to-code pretraining. While much of the information in a chart is provided by data table, the code that is used to render the table decides the visual layout (e.g., type of chart and orientation) and attributes (e.g., color) of the data. To test the importance of the chart-to-code pretraining component, we remove it in an ablated pretrained model and the model performance on ChartQA drops by 1.1% overall. The drop is mainly on the human set where more complex reasoning is required.

5.2 Fine-grained Analysis and Error Analysis

Fine-grained analysis. To understand the specific aspects of strengths and weaknesses of the models and breakdown the challenges into fine-grained categories, we sample 100 test examples from ChartQA (both augmented and human sets) for further analyses. Specifically, we summarize the challenges of ChartQA into three categories: (1) data extraction (where the model needs to parse a complex query with sophisticated coreference resolution or needs to read numbers when numbers are not explicitly written out), (2) math reasoning (where the model needs to perform one or multiple numerical operations such as min/max/sort/average/etc.), and (3) plot attributes (where the query asks about color/shape/location of specific objects/labels). We manually classify the 100 examples into the three categories and allow an instance to belong to multiple categories when the challenge is multifaceted. After excluding 7 annotation errors, we find 55.9% questions need complex data extraction, 45.2% involve math reasoning, and 7.5% concern plot attributes. We plot the per-category performance of PaLI (res. 588), Pix2Struct and MATCHA in Figure 2. Overall, all models perform the best on data extraction while math reasoning and plot attributes are more challenging. When compared across models, MATCHA improves Pix2Struct in every category and beats PaLI in both data extraction and math reasoning. However, for plot attributes, MATCHA lags behind PaLI. This is not significantly reflected in the overall ChartQA performance since plot attribute only concerns less than 10% of the examples.

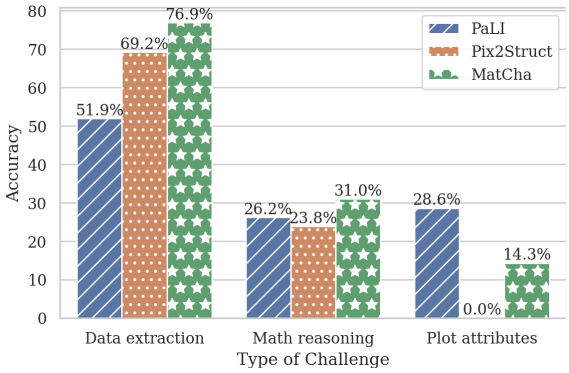
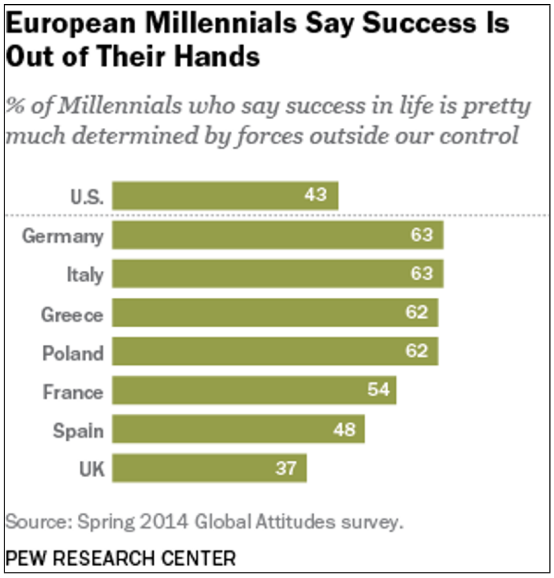


Figure 2: Fine-grained by-category performance comparison on ChartQA.

Error analysis. Similar to the fine-grained analysis, we sample 100 errors made by MATCHA on ChartQA test set and manually classify the 100 errors into the three categories. After exluding 21 annotation errors, we find 48.3% of the errors are related to math reasoning, 43.4% are related to data extraction, and 8.0% concern plot attributes. We conclude that math reasoning remains to be the major challenge even if MATCHA has improved its math reasoning capability compared with Pix2Struct and PaLI. We notice that MATCHA still struggles with sophisticated math reasoning questions or numerical computation that requires high precision. An example is shown in Appendix Table 8.

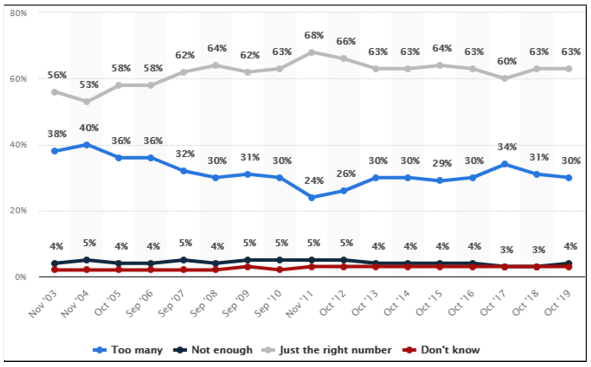
Case study. To concretely understand what type of questions MATCHA can do better than the baselines, we present several case studies. In Table 6, we show an example which requires computing average of multiple numbers. Besides MATCHA, PaLI and Pix2Struct's answers are far from the ground truth. In Table 7, we demonstrate an example that requires resolving complex coreference resolution of multiple data points. The model needs to accurately parse the query and find the referenced data points in the chart, then perform a simple numerical computation. MATCHA is the only model that gets the correct answer.

Besides cases where MATCHA succeeded, we



What is the average of last 4 countries' data?
PaLI: **40.94** Pix2Struct: **40.5** MATCHA: **50.5**

Table 6: An example that requires strong numerical reasoning skills. **Red** and **green** indicate correct and wrong answers respectively.

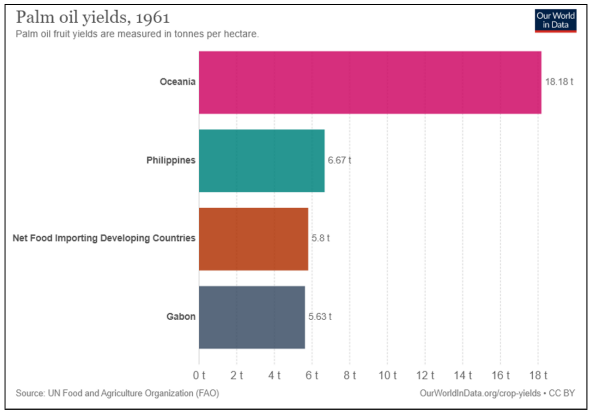


What percentage does 'don't known' and 'just the right number' make up for Oct'17?
PaLI: **10** Pix2Struct: **21** MATCHA: **63**

Table 7: An example that requires resolving both coreference resolution and math reasoning.

also present an example where all models have failed (Table 8). Questions which require very accurate numerical computation are still very challenging to MATCHA.

Continue pretraining Pix2Struct with its original objective. It is commonly known that BERT (Devlin et al., 2019) is undertrained and simply continuing training BERT with the same objective and on the same data for longer can slightly improve a model's performance (Liu et al., 2019). To understand whether such phenomenon persists for MATCHA and to what extent does continue



Is the sum of all last three places more than Oceania?
PaLI: **Yes** Pix2Struct: **Yes** MATCHA: **Yes**

Table 8: An error made by all models including MATCHA which requires very accurate numerical computation. The answer should be 'No' since 6.67+5.8+5.63=18.1<18.18.

pretraining on Pix2Struct screenshot parsing task would improve the model's performance, we continue pretraining Pix2Struct with its original objective and data for 50k steps. We found that continue pretraining indeed improves Pix2Struct's performance (56.0→57.0 on ChartQA) but is to a much less extent without using the MATCHA pretraining components (improving from 56.0 to 64.2).

6 Conclusion

We have proposed a pretraining method MATCHA for visual language tasks. MATCHA injects chart understanding and reasoning knowledge to an image-to-text transformer model by learning to (1) predict the underlying data tables and code given chart images and (2) decode the answers of math questions (rendered in the form of images). MATCHA establishes new SOTA on 5 out of 6 setups across three chart domain benchmarks covering both QA and summarization tasks. On visual language tasks beyond the chart domain (e.g., textbook QA and DocVQA), MATCHA improves upon Pix2Struct, indicating that the learned knowledge in MATCHA pretraining can be transferred outside of the pretraining domain. We conduct comprehensive ablation studies to identify the actual impact of each pretraining component and task and find that chart derendering is essential for extractive questions while math pretraining is important for queries that requires complex reasoning.

Limitations

Though we have injected math reasoning skills to MATCHA, error analysis shows that there is still room for improvement on queries requiring complex reasoning. Besides, it remains debatable whether doing math calculation in weight space in a purely end-to-end manner is the most promising path forward.[9]

Besides math reasoning, Figure 2 shows that plot attributes is an area where MATCHA underperforms PaLI. We conjecture that it is due to MATCHA's lack of massive scale grounded image-text pretraining with rich semantics (which PaLI has using web-scale image-text pairs). While chart-to-code pretraining provides certain level of plot attribute grounding, such plot features are mostly using default options in plotting packages but not explicitly written out in code.

In terms of experimental setup, the reported number is result of a single run. Pretraining is extremely costly especially when there exists more than twenty ablation setups and downstream evaluation tasks. We have collected pretraining and evaluation data points from multiple aspects on various scenarios to verify the robustness of MATCHA. However, we do acknowledge that the paper can benefit from reporting multiple runs given sufficient compute.

Last but not least, it is also worth noting that visual language is an umbrella term. There are other visual language systems beyond the ones discussed in this paper. As an example, comics/manga have their distinct visual lexicon or even grammars (Cohn, 2013).

Ethics Statement

To the best of our knowledge, MATCHA has not been trained on sensitive private information and should be of low risk to generate harmful contents. All pretraining and finetuning data are either synthetically created using rules or publicly available data on the web with appropriate permissive licenses.

References

Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. 2023. Reading and reasoning over chart images for evidence-based automated fact-checking. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 399–414, Dubrovnik, Croatia. Association for Computational Linguistics.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. 2023. Pali: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, et al. 2023. Binding language models in symbolic languages. In *The Eleventh International Conference on Learning Representations*.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR.

Neil Cohn. 2013. *The Visual Language of Comics: Introduction to the Structure and Cognition of Sequential Images*. A&C Black.

Brian L. Davis, B. Morse, Bryan Price, Chris Tensmeyer, Curtis Wigington, and Vlad I. Morariu. 2023. End-to-end document recognition and understanding with dessurt. In *Computer Vision – ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, page 280–296, Berlin, Heidelberg. Springer-Verlag.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[9] See recent works that combine LLMs with calculators (Wei et al., 2022) or compilers/program executors (Cheng et al., 2023; Chen et al., 2022; Gao et al., 2022).

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 281–296, Online. Association for Computational Linguistics.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. PAL: Program-aided language models. *arXiv preprint arXiv:2211.10435*.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Robert E Horn. 1998. Visual language. *MacroVu Inc. Washington*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. LayoutLMv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 4083–4091, New York, NY, USA. Association for Computing Machinery.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. DVQA: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. FigureQA: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*.

Shankar Kantharaj, Rixie Tiffany Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. Chart-to-text: A large-scale benchmark for chart summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4005–4023, Dublin, Ireland. Association for Computational Linguistics.

Jihyung Kil, Soravit Changpinyo, Xi Chen, Hexiang Hu, Sebastian Goodman, Wei-Lun Chao, and Radu Soricut. 2022. PreSTU: Pre-training for scene-text understanding. *arXiv preprint arXiv:2209.05534*.

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

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning*.

Matan Levy, Rami Ben-Ari, and Dani Lischinski. 2022. Classification-regression for chart comprehension. In *European Conference on Computer Vision*, pages 469–484. Springer.

Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. 2021. Visually grounded reasoning across languages and cultures. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10467–10485, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2023. DePlot: One-shot visual language reasoning by plot-to-table translation. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and

logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020. PlotQA: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1527–1536.

Xinyu Pi, Qian Liu, Bei Chen, Morteza Ziyadi, Zeqi Lin, Qiang Fu, Yan Gao, Jian-Guang Lou, and Weizhu Chen. 2022. Reasoning like program executors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 761–779, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. 2017. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223, Vancouver, Canada. Association for Computational Linguistics.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Yuhuai Wu, Felix Li, and Percy Liang. 2022. Insights into pre-training via simpler synthetic tasks. In *Advances in Neural Information Processing Systems*.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200.

A More Details on Datasets Used

Chart-table pairs from the web. The data was originally collected by Masry et al. (2022) and came from the below four sources:

- Statista: `www.statista.com`
- Pew: `www.pewresearch.org`
- Our World in Data: `ourworldindata.org`
- OECD: `www.oecd.org`

Modules of MATH questions included. We exclude overly complex math questions and only select the basic modules that would help with numerical reasoning. They are from the two areas of Arithmetic and Comparison. The individual modules included are

- Arithmetic
  - `add_or_sub`
  - `add_sub_multiple`
  - `div`
  - `mixed`
  - `mul`
  - `mul_div_multiple`
- Comparison
  - `closest`
  - `closest_composed`
  - `kth_biggest`
  - `kth_biggest_composed`
  - `pair`
  - `pair_composed`
  - `sort`
  - `sort_composed`

Please see Saxton et al. (2019) for detailed descriptions about each module and how they are generated.

B Details of Baselines

We introduce below the details of the baselines used in Table 3.

T5 is an encode-decoder Transformer model proposed by Raffel et al. (2020). The baseline model T5 takes the concatenation of a linearized table (and a query, when the task is QA) as input, and aims to decode the target (answer or summarization). When the gold table is availible, the gold table is used as the input and the chart image is not used directly. VL-T5 proposed by Cho et al. (2021) is similar to T5 but also takes a visual input (i.e., the chart image) on the encoder side. VisionTaPas (Masry et al., 2022) is modified from TaPas (Herzig et al., 2020) to incorporate the visual modality by adding a ViT model (Dosovitskiy et al., 2021) and

cross-modal fusion layers. T5-OCR, VL-T5-OCR, and VisionTaPas-OCR are the same model as T5, VL-T5, and VisionTaPas, respectively. However, they do not assume the existence of gold table but use an OCR-based system to extract the data table from the chart image. The above mentioned models and their performance numbers are all extracted from Masry et al. (2022) and Kantharaj et al. (2022). Please see the original paper for more details. Classification - Regression Chart Transformer (CRCT) (Levy et al., 2022) is the best performing model on PlotQA according to the PlotQA benchmark on paperswithcode.com. It uses a detector that extracts all textual and visual elements of chart then processes these elements with a multimodal Transformer.