# Chart-based Reasoning: Transferring Capabilities from LLMs to VLMs

**Victor Cărbune*** **Hassan Mansoor** **Fangyu Liu** **Rahul Aralikatte**
**Gilles Baechler** **Jindong Chen** **Abhanshu Sharma**
Google Research

## Abstract

Vision-language models (VLMs) are achieving increasingly strong performance on multimodal tasks. However, reasoning capabilities remain limited particularly for smaller VLMs, while those of large-language models (LLMs) have seen numerous improvements. We propose a technique to transfer capabilities from LLMs to VLMs. On the recently introduced ChartQA, our method obtains state-of-the-art performance when applied on the PaLI3-5B VLM by Chen et al. (2023c), while also enabling much better performance on PlotQA and FigureQA.

We first improve the chart representation by continuing the pre-training stage using an improved version of the chart-to-table translation task by Liu et al. (2023a). We then propose constructing a 20x larger dataset than the original training set. To improve general reasoning capabilities and improve numerical operations, we synthesize reasoning traces using the table representation of charts. Lastly, our model is fine-tuned using the multitask loss introduced by Hsieh et al. (2023).

Our variant ChartPaLI-5B outperforms even 10x larger models such as PaLIX-55B without using an upstream OCR system, while keeping inference time constant compared to the PaLI3-5B baseline. When rationales are further refined with a simple program-of-thought prompt (Chen et al., 2023a), our model outperforms the recently introduced Gemini Ultra and GPT-4V.

## 1 Introduction

Visual language, where text and images work together to deliver information, can be expressed through charts, plots, and diagrams. Multimodal reasoning within this context is challenging, as it involves linking visual properties (like color, line style, and positioning) with textual content (such as legends and units).

Many recent advances of vision-language models (VLMs) come from techniques enabling better representations (Dosovitskiy et al., 2021; Lee et al., 2023), giving the model the ability to understand core elements of the image, a necessary building block for basic reasoning. However, complex reasoning capabilities which combine the core representation of the image with semantic understanding of a question to provide an answer, have been rather limited. Models oftentimes are not able to contextually combine image and text representations. One technique that improves reasoning capabilities in large-language models (LLMs) includes in-context learning for eliciting reasoning such as chain-of-thought prompting (Wei et al., 2023), decomposing tasks (Zhou et al., 2023) or composing stored facts in weights (Press et al., 2023). Fine-tuning on datasets with rationales (Magister et al., 2023; Hsieh et al., 2023) has been shown to be effective for smaller models. In this work, we tackle improving reasoning capabilities in VLMs through better learned image representations, followed by fine-tuning on synthetic datasets with reasoning traces generated by more capable LLMs. We also explore a hybrid online setup for numerical reasoning refinements.

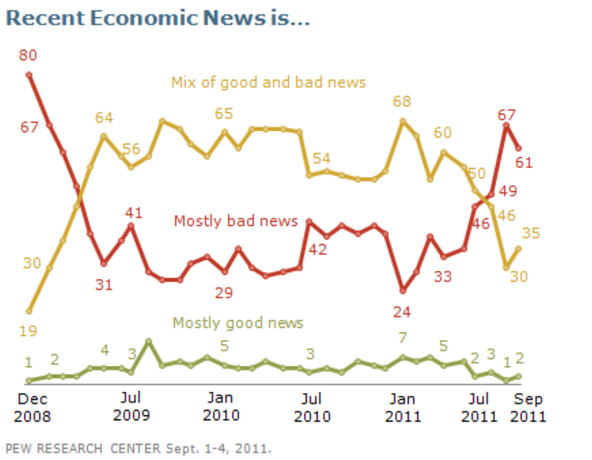
We empirically show that this indeed improves performance through experiments on ChartQA (Masry et al., 2022). Visual-question answering on charts quantifies the ability of a VLM to reason using complex information presented. Oftentimes answering the question requires implicit or explicit information extraction, followed by intermediate grouping or computations using the extracted information, and reasoning with the final quantities, as shown in Figure 1.

Vision-language models (VLMs) such as PaLI-X and PaLI-3 are hybrid model architectures which use a vision and a language backbone to solve visual tasks (Chen et al., 2023b,c). The training recipe typically involves a pre-training stage fo-

* Correspondence to: vcarbune@google.com



*Findings of the Association for Computational Linguistics: NAACL 2024*, pages 989–1004
June 16-21, 2024 




**Question:** What's the difference between the highest value of the red line and the lowest value of the green line? **Answer**: 79

Fig. 1: Example from the ChartQA validation set.

cused on learning a good internal representation, followed by a downstream fine-tuning stage. Chen et al. (2023c) note that PaLI-3 falls behind PaLI-X on ChartQA likely due to its limited reasoning capabilities. Results presented in this work suggest that the lack of a pre-training task for learning better chart representations, as done in Liu et al. (2023b), may be another reason.
Enhancing the reasoning capabilities of large language models (LLMs) such as PaLM-2 (Anil et al., 2023) or GPT-4 (OpenAI, 2023) is a very active research area. While reasoning is considered an emerging property with scale (Wei et al., 2022), Press et al. (2023) argue that simply scaling only enables better memorization of knowledge and does not enable composing multiple stored facts into an answer. On the other hand, prompting techniques enacting complex reasoning on downstream tasks have been shown to be very effective (Wei et al., 2023) (Zhou et al., 2023).

Transferring reasoning capabilities from large to small models enables reducing serving costs, while increasing task performance. Hsieh et al. (2023) have introduced an effective multi-task framework which enable small models to outperform their much larger counterparts using less data. They do so by leveraging rationale generation as a separate task, instead of more standard distillation approaches, which first infer the rationale, followed by the answer (Magister et al., 2023). We apply this framework for the first time on multimodal tasks.

**Contributions** Our main results can be summarized as follows: **(i)** we introduce an efficient recipe consisting of a pre-training task and fine-tuning task with synthetic datasets using a multi-task setup for improving reasoning capabilities, **(ii)** we obtain SoTA performance by significantly improving PaLI-3 performance on the ChartQA benchmark with our recipe and using 10x less parameters than prior work, **(iii)** we perform numerous ablation experiments quantifying the impact of the techniques used in our recipe.

The remainder of this paper is structured as follows. Section 2 describes related work, followed by Section 3 which introduces the construction of the training datasets. Section 4 illustrates our novel pre-training and fine-tuning recipe, followed by Section 5 describing the experimental setup and main results. Lastly, Section 8 delivers a conclusion and recommendation for future work, followed by Section 9 where we acknowledge limitations of the current work.

## 2 Related Work

**VLM landscape** Vision-language models usually combine a vision backbone with a language backbone. Frequently it is a Vision Transformer (ViT) (Dosovitskiy et al., 2021) coupled with a Large Language Model via an encoder-decoder (Chen et al., 2023b) or decoder-only (Alayrac et al., 2022) architecture. More recently, models such as Fuyu-8B (Bavishi et al., 2023) explore projecting the image directly through the language backbone. In this work we extend PaLI-3, an encoder-decoder architecture with ViT-3B as vision and UL2-2B as language backbones. We refer the reader to Chen et al. (2023c) for a complete overview. PaLI-3 is a SoTA model and hence we decided to build on top of it to further focus on improving the results with our methods.

**Existing approaches for chart understanding** The task of answering questions on charts is, alongside documents and infographics, part of a broader set of tasks commonly referred to *visually-situated language understanding*, where text and image cannot be treated separately (Lee et al., 2023). Fine-tuned models on downstream ChartQA include PaLI-3 (Chen et al., 2023c), MatCha (Liu et al., 2023b) and UniChart (Masry et al., 2023). Among these, UniChart takes the most similar approach to ours, pre-training a chart image encoder as vision backbone and BART decoder (Lewis et al., 2019) as language backbone. Alternatively, Liu et al. (2023a) took the approach of decomposing question-answering into first translating the chart into a table, then querying an LLM in a plug-and-play fashion. Here our main focus is on fine-tuned self-contained models, however we show that a simple refinement using a much larger LLM, continues to improve performance as well.

**The role of upstream OCR systems** A chart usually has an underlying equivalent tabular representation of the data. However, decoding the tabular representation remains a challenging problem. Alternatively, charts can be passed through an OCR system to extract an unstructured text representation of the image. (Luo et al., 2021) combine chart-specific extraction logic with an OCR system to extract key information from the charts. As intuitively expected, usually the use of an OCR system improves downstream quality. In this work, we assume the model only has access to the chart image.

**Improving chart reasoning with synthetic data** Having the pre-training mixture specialize on chart tasks is effective (Liu et al., 2023b). We further extend the *chart derendering* task, which translates charts to code or to table. Similar to our approach, Methani et al. (2020) and Masry et al. (2023) have made use of programmatic templates to a synthesize complex QA pairs. However, instead of using an LLM to generate chart summaries as in Masry et al. (2023), here we use it to generate additional QA pairs with rationales. These generated examples together with synthetic programmatic examples are key in the pre-training and fine-tune stages of our model.

# 3 Dataset

## 3.1 Brief description of ChartQA

ChartQA is one of the widely adopted visual question-answering benchmarks for reasoning capabilities of VLMs.

The standard ChartQA benchmark has two components: (a) human set and (b) augmented generated set. The augmented set has been machine generated and is more simplistic in nature than the human set.

The charts in the dataset come from four sources (Statista, Pew, Our World in Data and OECD). Gold tables are available for all sources, except for Pew, where the tables are inferred with ChartOCR model (Luo et al., 2021). Although we observed mistakes in inferred tables, our method seems to be fairly resilient to them.

## 3.2 Synthetic Generation Methods

In this work, we use LLMs to synthesize additional examples paired with rationales generated using chain-of-thought prompting. We use the tabular representation of charts present in the training set as a way to mediate the lack of vision input into LLMs.

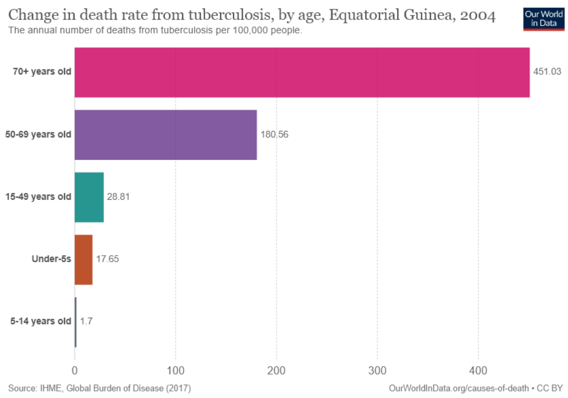
The data we synthesize increases the diversity of the original training set, especially with examples that require extracting multiple quantities from the chart and perform reasoning using them.

We combine two approaches that focus on this type of examples, specifically we use a LLM for synthesizing *rationale generation* and *extra question answer* pairs. We also use a programmatic approach for generating *arithmetic* question answer pairs.

**Rationale Generation** We augment the original training set with synthetic explanations on why an answer is reached. We achieve this by using PaLM 2-S to predict a $\mathbf{rationale}$ on an input tuple of $(\mathbf{table}, \mathbf{question}, \mathbf{answer})$ with a 4-shot prompt, as illustrated in Figure 4. We refer to this set as *ChartQA-Rationale-S*.

By requesting the model to provide justifications for ground truth answers, which are typically accurate, we witness a significant reduction in hallucinations. A notable exception is when the answer itself is wrong, which happens more frequently for the ChartQA augmented set than the human set. However, we did not perform a detailed investigation of this aspect in the generated training sets. An instance of the generated rationale can be seen in Figure 2.

**ExtraQA Generation** We hypothesize that the original training set is too small to contain enough diversity in the examples to enable solving more complex QA questions such as the ones present in the human validation set. Therefore we used a 1-shot prompt illustrated in Figure 5 to generate additional examples covering types of errors we identify by examining the model performance on the validation set. The prompt is adapted from the one used in (Liu et al., 2023a). An example of a generated sample can be seen in Figure 7. We used both PaLM 2-S and PaLM 2-L to generate the examples and refer to the respective datasets as *ChartQA-ExtraQAR-S/L*. We perform only lightweight filtering of generated examples that deviate from the imposed structure. If we cannot parse from the LLM




**Question:** "Find the difference between the largest value and the median of all values?"
**Table:** "TITLE | Change in death rate from tuberculosis, by age, Equatorial Guinea, 2004\nCountry | Change in death rate from tuberculosis, by age, Equatorial Guinea, 2004\n70+ years old | 451.03\n50-69 years old | 180.56\n15-49 years old | 28.81\nUnder-5s | 17.65\n5-14 years old | 1.7"
**Answer:** 422.22
**Rationale:** "The table shows the change in death rate from tuberculosis by age in Equatorial Guinea in 2004. The largest value is 451.03 and the median is 17.65. The difference between the largest value and the median is 422.22."

Fig. 2: *ChartQA-Rationale-S*: For each example of the original training set, we synthesize a rational based on the table, the question and the answer.

response all three elements, we simply drop the example. However, we do not verify the generated examples for hallucinations, fluency or perform any other model-based verification.

**ArithmeticQA Generation** It is well known that large language models have difficulties in performing arithmetic computations accurately. For ChartQA, this is particularly exacerbated by the fact that the small training dataset is adequate for the specifics of the arithmetic questions one can have for charts (as represented by the test set). We programmatically create examples which either require numeric reasoning or a comparative analysis of multiple chart elements. Examples are illustrated in Figure 8 and Figure 9. We abstracted the questions into templates and used a fixed set of mathematical operations such as median, max, min etc. For each template we created a rationale to teach the model a plan to solve the arithmetic problems. For example, computing the mean requires first looking up the values, then adding them up and finally dividing the value by the total. For each type of arithmetic we created multiple templates both for the questions and rationales. The source data we used are only the ChartQA human examples, using the available tables. The type of questions and their count can be found in Table 1.

| Question Type | Count # |
|---|---|
| Mean | 235K |
| Subtraction | 90K |
| Other | 32K |
| Total | 357K |

Table 1: Examples are mostly means or subtractions.

### 3.3 Resulting Dataset

The resulting dataset is roughly 20x larger and is described in Table 2, with further details on the statistics of the dataset in Section D. Sampling was done using greedy decoding with temperature $\tau = 0$. We used the augmented and human sets to generate examples.

**PaLM 2-S vs. 2-L** The same prompt was used for all examples in the synthetic dataset. We note that using samples from both LLMs improves performance, but ablation studies do not indicate one is better than the other. We hypothesize that diversity matters more than model size, but we have not investigated sampling strategies.

## 4 Method

Our work builds on top of PaLI-3 architecture and pre-training recipe, which consists of two backbones, a Vision Transformer ViT-2B and Text Encoder-Decoder UL2-3B. Our starting point is the recipe described by Chen et al. (2023c). The uni-modal pre-training stage trains the vision encoder using contrastive loss through the SigLIP loss, while the language encoder-decoder is pre-trained using the UL2 loss. Both backbones are pre-trained jointly using a multi-modal stage. Lastly the resolution increase stage enables the vision encoder backbone to work with 812x812 resolution images. We continue pre-training using this checkpoint.

### 4.1 Pre-training: Chart2Table Mixture

Extending the work done by Liu et al. (2023a), we use a chart-to-table dataset mixture to continue pre-training with the ViT backbone unfrozen, which facilitates learning an internal representation of the chart. We do not explicitly use the tabular conversion further downstream.

**Dataset** For learning this representation, we combine several chart-to-table derendering tasks into a mixture: (1) synthetic chart-to-table data similar to the synthetic mixture introduced by Liu et al.

| Dataset | Hum # | Aug # | Question type # | Total | Rate # |
|---|---|---|---|---|---|
| ChartQA-Rationale-S | 7398 | 20901 | R [13%], V [11%], C [43%], B [33%] | 28.3K | 15% |
| ChartQA-ExtraQAR-S | 23261 | 69433 | R [57%], C [43%] | 92.7K | 15% |
| ChartQA-ExtraQAR-L | 16388 | 50468 | R [60%], C [40%] | 66.9K | 30% |
| ChartQA-ArithmQAR | 357000 | - | C [100%] | 357.0K | 40% |
| ChartQA-Synth (Total) | | | | **544.9K** | |

Table 2: Overview of the synthetic dataset, which is 20x larger than the original one. The suffix denotes the size of the PaLM 2 model used. The rate refers to the final mixture. Categorization of question types are from (Masry et al., 2022), namely **R**etrieval, **V**isual, **C**ompositional or **B**oth visual and compositional.

(2023a). We traverse different combinations of plotting options in matplotlib and seaborn to randomly plot tables from Wikipedia into charts of different layouts. (2) the chart-to-table mixture introduced by Masry et al. (2023). (3) The chart-table pairs from the train set of DVQA (Kafle et al., 2018). (4) The chart-table pairs from the train set of TaTA (Gehrmann et al., 2022). (5) The chart-table pairs introduced in Benetech - Making Chart Accessible Kaggle challenge[1]. A complete listing of data source, sampling weight, and number of examples is shown in Table 3.

| Component | Rate | Size |
|---|---|---|
| Synthetic | 44.0% | 1.2M |
| UniChart | 39.5% | 612K |
| DVQA | 3.2% | 200K |
| ChartQA | 3.2% | 22K |
| TaTa | 3.2% | 6.7K |
| Chart2Text | 3.2% | 24K |
| Benetech Challenge | 3.2% | 21K |
| PlotQA | 0.5% | 224K |
| Total | | **2.37M** |

Table 3: Pre-training datasets for learning chart representations include examples from numerous tasks that have paired chart images with table representations.

The existing table representation is used as is from the datasets, or, as described earlier, for a small fraction, tables are created programmatically. Tables are also normalized to a standardized format.

### 4.2 Fine-tuning: Multi-task Loss

After the pre-training stage which enables the ViT backbone to work better with charts, we use the synthetic data to fine-tune the model for the downstream task. We investigate two ways of incorporating the rationales available in the extended dataset.

[1]`https://www.kaggle.com/competitions/benetech-making-graphs-accessible`

The first one is by changing the task target from *answer* to *rationale, answer*. This has been shown to be effective in (Magister et al., 2023). We refer to this approach as **single-task setup**. However, it requires increased inference time by predicting the rationale, together with increased sequence length during training. The unintended side effect of training to predict jointly rationales and answers is that rationale tokens become equally important as the answer tokens.

The second one is inspired by Hsieh et al. (2023) which addresses both concerns by constructing a **multi-task setup** where the answer and rationale are treated as independent tasks. This can be done using different prefixes similar to T5 (Raffel et al., 2023), such as *"Rationale:"* and *"Question:"*. The training loss balances the strength between the two tasks using a hyper-parameter $\lambda$:

$$\mathbf{Loss} = (\mathbf{1} - \lambda)\mathbf{Loss_{ans}} + \lambda\mathbf{Loss_{rat}}$$

Our experiments are the first application of this setup for a multimodal task. We further confirm the observation from text domains that not only inference time remains constant, but quality also improves.

## 5 Experiments

We describe the general learning hyper-parameters for the pre-training and fine-tuning stages, followed by interpretation of the results.

### 5.1 Setup

**Pre-training** We continue pre-training the PaLI-3 model with ViT unfrozen on the Chart2Table data mixture for `train_steps=6K`, `batch_size=256` with `learning_rate=5e-3` with normalized square root decay using `decay_factor=2e-6` and `dropout_rate=0.1`.

**Fine-tuning** We then freeze the ViT encoder and continue fine-tuning on the synthetic ChartQA dataset for `train_steps=5K`, `batch_size=256` with `learning_rate=1e-3` with linear decay using `decay_factor=1e-4` using `dropout_rate=0.1`.

**Multitask** We use $\lambda = \mathbf{0.5}$ and we do not find significant differences when using other values.

### 5.2 Results on ChartQA

We validate the effectiveness of the different techniques by reporting the downstream task performance on the ChartQA test set. All following experiments are on PaLI-3.

**Pre-training** Continuing the pre-training stage for the PaLI-3 model using the Chart2Table mixture enables learning a better general representation of the charts. We intuitively expect that this better representation enables the model to more accurately identify quantities on the images. Indeed, we confirm this first through the results reported in Table 4. Later, as we scale the dataset size, we show that this continues to play an important role.

| Pre-training Strategy | ChartQA (RA%) | | |
|---|---|---|---|
| | Avg. | Hum. | Aug. |
| Original PT (Chen et al., 2023c) | 70.00 | - | - |
| Chart2Table PT (our run) | **70.84** | 48.96 | 92.72 |

Table 4: PaLI-3 performance on ChartQA slightly increases with our chart-to-table pre-training phase.

As expected, the increase is predominantly in the augmented set, given that the pre-training mixture is constructed synthetically as well.

**Singletask vs. Multitask** We first study the effect of introducing rationales only using the *ChartQA-Rationale-S*. This only adds rationales to the original ChartQA dataset.

When using the rationales in singletask setup the performance difference is not significant compared to not using them. However, when used in the multitask setup, we note a quality improvement, particularly noticeable in the more difficult human-set. We refer to the former as *Singletask-Rationale* and to the latter as *Multitask-Rationale* in Table 5.

| Fine-tuning setup | ChartQA (RA%) | | |
|---|---|---|---|
| | Avg. | Hum. | Aug. |
| C2T PT + Singletask-Rationale | 70.80 | 49.36 | 92.24 |
| C2T PT + Multitask-Rationale | **71.72** | 50.72 | 92.72 |

Table 5: Multitask performance stands out compared to Singletask on the more difficult human-written set.

We hypothesize that the improvement comes from better use of the rationales, guiding the model to internally produce a form of reasoning before producing the final answer. This is done without paying the cost predicting the rationales tokens.

**Learning with augmented dataset** We use the ChartQA-Synth dataset from Table 2 for studying the extent to which we can transfer reasoning capabilities from PaLM-2 to PaLI-3.

We perform an ablation experiment to understand the role of the extra questions, rationales and pre-training stage and report our results in Table 6.

We denote experiments using the original pre-trained checkpoint as *Orig PT* and on the further pre-trained checkpoint with chart-to-table translation as *C2T*. We report a clear improvement, further strengthening our observation that internal representation plays an important role.

| Fine-tuning Setup | ChartQA (RA%) | | |
|---|---|---|---|
| | Avg. | Hum. | Aug. |
| Orig PT + Singletask-ExtraQAR | 72.43 | 53.20 | 91.67 |
| Orig PT + Multitask-ExtraQAR | 73.15 | 55.20 | 91.10 |
| C2T PT + ExtraQA (w/o Rationale) | 74.67 | 56.39 | 92.96 |
| C2T PT + Singletask-ExtraQAR | 75.16 | 55.84 | **94.48** |
| C2T PT + Multitask-ExtraQAR | 75.36 | 56.80 | 93.92 |
| C2T PT + Singletask-ChartQA-Synth | 76.60 | 59.04 | 94.16 |
| C2T PT + Multitask-ChartQA-Synth | **77.28** | **60.88** | 93.68 |

Table 6: Ablation results confirm the importance of each step in our recipe. *ChartQA-Synth* is the mixture described in Table 2

We ran an experiment without rationales, but with the entire synthetically generated QA pairs. We note that the increase in examples ends up improving over the original ChartQA performance reported in Table 4. However, the use of rationales continues to improve quality for both singletask and multitask setups. We observe that in high-data regimes, there is no longer a significant difference between the two.

Given the neutral impact of the multi-task setup at inference time, paired with slightly improved performance on the human-written queries of ChartQA, multi-task is the preferred option in practice. Further, we refer to the best performing fine-tuned setup in Table 6 as **ChartPaLI-5B**.

### 5.3 Results on FigureQA and PlotQA

ChartQA is currently the most challenging benchmark. To prove that our method is general, we investigate performance on related chart understanding tasks, FigureQA () and PlotQA (). We study 3 operation regimes: (i) **zero-shot**: no task-specific pre-training or fine-tuning, (ii) **quick adaptation**: 1K fine-tuning steps and (iii) **convergence**: 5K fine-tuning steps. We report relaxed accuracy on 10K examples from validation set for FigureQA (ref. Table 8 and from test set from PlotQA (ref. Table 9).

| **Model** | FigureQA RA% (v1 ǀ v2) | | |
|---|---|---|---|
| | **ZShot** | **Quick** | **Conv** |
| PaLI-3 (original) | 41.9 ǀ 42.4 | 57.2 ǀ 58.1 | 89.9 ǀ 89.3 |
| ChartPaLI-5B | **51.0 ǀ 51.2** | **92.7 ǀ 93.0** | **96.3 ǀ 96.2** |

Table 8: ChartPaLI-3 exhibits strong generalization on FigureQA task, for which no examples are present in pre-training or fine-tuning

For PlotQA, images from the training subset are present in our pre-training mixture, while validation and test subset images are not. Therefore, we do not study zero-shot performance, as training images would give an unfair advantage.

| **Model** | PlotQA RA% (v1 ǀ v2) | |
|---|---|---|
| | **Quick adapt.** | **Convergence** |
| PaLI-3 (original) | 62.0 ǀ 15.7 | 71.5 ǀ 23.6 |
| ChartPaLI-5B | **79.1 ǀ 53.3** | **86.0 ǀ 70.7** |

Table 9: ChartPaLI-3 exhibits strong generalization on FigureQA task, for which no examples are present in pre-training or fine-tuning

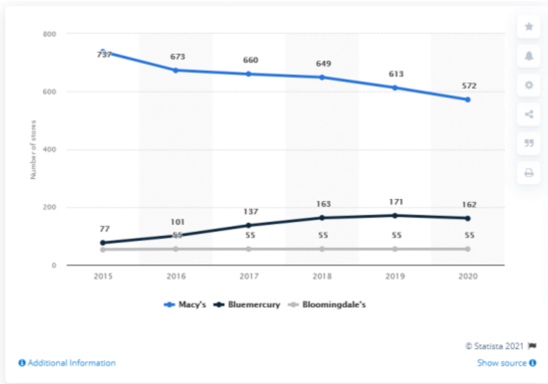
ChartPaLI-5B outperforms PaLI-3 in all operation regimes. In general, our recipe significantly increases chart understanding performance when running only a few quick adaptation steps.

In particular we report SoTA performance regime for FigureQA (roughly 96%+) and the very strong relative performance on the difficult PlotQA v2 (roughly +47.1% at convergence time).

### 5.4 Errors and Challenges

To understand the effect of our method and investigate further opportunities for improvements, we manually looked at predictions on the ChartQA validation set. We compared baseline PaLI-3 model outputs with the model fine-tuned with our recipe and share our observations below. We report our findings below.

**General** The model predicts the rationale[2] or the answer, depending on the task prefix. Because the answer is not conditioned on the rationale, it can differ. One general improvement area we note is the ability to extract necessary intermediate quantities (Fig. 11) and operate with them (Fig. 12).




**Question:** What is the total of Macy's, Bloomingdale's and Bluemercury brand stores worldwide in the 2019?
**Answer** [baseline]: 1122
**Answer** [ours]: 832 *[inexact, but correct because it's within 5% of ground truth]*
**Rationale** [ours]: The table shows the number of Macy's, Bloomingdale's and Bluemercury brand stores worldwide in the years 2015, 2016, 2017, 2018, 2019 and 2020. In 2019, **Macy's had 613** brand stores, **Bloomingdale's had 55** brand stores and **Bluemercury had 171** brand stores. Therefore, the total number of Macy's, Bloomingdale's and Bluemercury brand stores worldwide in the 2019 is 613 + 55 + 171 = 824.

Fig. 3: Correct numeric approximations on answers.

**Numerical reasoning** Despite improvements, computation of mathematical expressions continues to be very challenging. The rationales correctly extract (Fig. 3) or infer chart values when missing (Fig. 13), however the computed value is frequently incorrect. This does not always prevent the final answer to be correct (Fig. 15). This seems in line with observations by Wang et al. (2023a), who also conclude that corruption of the chain-of-thought reasoning trace does not always degrade the final answer. Due to the frequency of this numeric computation error, we explore a simple refining technique in Section 5.5.

[2]Although the table is not used during inference, the rationales contain the word *table* due to its use in prompts.

| **Fine-tuned VLMs** (up to 55B) | Source | **ChartQA (RA%)** |
|---|---|---|
| Fuyu-8B | our eval, (Bavishi et al., 2023) | 42.1 |
| Pix2Struct-1.3B | (Lee et al., 2023) | 58.6 |
| MatCha-300M | (Liu et al., 2023b) | 64.2 |
| UniChart-201M | (Masry et al., 2023) | 66.2 |
| ChartLlama-13B | (Han et al., 2023) | 69.6 |
| PaLI-5B | (Chen et al., 2023c) | 70.0 |
| PaLI-55B | (Chen et al., 2023b) | 70.9 |
| PaLI-55B (Soft Mixture of Low-rank Experts) | (Wu et al., 2023) | 73.8 |
| ChartPaLI-5B | **our work** | **77.3** |
| **Hybrid VLMs/LLMs** (undisclosed size) | | |
| GPT-4V [4-shot with CoT] | (OpenAI, 2023) | 78.5 |
| DePlot-300M + FlanPaLM + Codex with PoT SC | (Liu et al., 2023a) | 79.3 |
| Gemini Ultra [0-shot] | (Gemini Team, Google, 2023) | 80.8 |
| ChartPaLI-5B + PaLM 2-S PoT SC @ 5 | **our work** | **81.3** |

Table 7: State-of-the-art performance among fine-tuned VLMs on ChartQA benchmark.

**Color reasoning** Our synthetic data does not have color metadata, as only the table was used in the generation process. Therefore the model continues to struggle when the reasoning trace requies working with colors (Fig. 10). Thus, this is an area worth of investigating next and has applicability well beyond the specifics of chart understanding.

**Complex reasoning** Reasoning about multiple values and checking for a matching condition which requires arithmetic computations is another example of a remaining difficult task (Fig.14, Fig.16). The increased complexity stemming from internal inability of VLMs to perform numeric operations paired with enumerating chart elements through semantic descriptions is likely fairly difficult to achieve without the use of external tools.

**Task leakage** Due to the training methodology, we observe that when conditioned with the *Question* task prefix, the model may behave similarly as to when *Rationale* prefix is used. Sometimes, instead of directly outputting an answer, the model may generate a longer explanation that resembles a rationale or a fragment of rationale.

## 5.5 Refinement with Program of Thoughts

Despite the improved ability to construct numeric equations using the required values on the charts (Fig. 3), the exact numeric computation continues to be wrong. This is unsurprising, since both the visual and the language backbone treat numbers as tokens. Making the problem worse, the character sequence forming a number may be split and encoded in arbitrary chunks. Chen et al. (2023a) have proposed replacing chain-of-thoughts (CoT) prompting with program-of-thoughts (PoT) to enable delegation of the arithmetic computation to a program interpreter. This has previously been explored by Liu et al. (2023a), however in a much more computationally involved setup than the one we describe further.

Through our fine-tuning approach, both single-task and multitask setups can be used produce CoT rationales for which an LLM prompted with PoT can write the equivalent code for performing the numeric computation.

We take the approach of using a simple 4-shot prompt (Fig. 6) constructed on the validation set to generate code using PaLM 2-S for performing the numeric computation that is present in a rationale. We run this online refinement, only if the rationale contains an arithmetic operator ('+', '-', '/' or '*').

Self-consistency is an effective way to improve chain-of-thoughts rationales by selecting an answer with majority voting from a pool of sampled rationales (Wang et al., 2023b). We apply this approach, by sampling with temperature $\tau_{Rat} = 0.4$ and generate $N = 5$ rationales that are then refined with PaLM 2-S using temperature $\tau_{Ref} = 0.0$.

| **Setup** | ChartQA (RA%) | | |
|---|---|---|---|
| | **Avg.** | **Hum.** | **Aug.** |
| ChartPaLI-5B (from Table 6) | 77.28 | 60.88 | 93.68 |
| ChartPaLI-5B + PaLM 2-S PoT | 80.80 | 67.92 | 93.68 |
| ChartPaLI-5B + PaLM 2-S PoT SC @ 5 | **81.32** | **68.96** | 93.68 |

Table 10: PoT refinement improves performance on the human set, while not affecting the augmented set.

The results presented in Table 10 highlight the utility of the method, particularly with K=5 for self-consistency. They also highlight the simplicity of the augmented set compared to the human set, for which the refinement does not have an impact. Either the augmented set contains no arithmetic computations or they are simple enough for the fine-tuned VLM to already get right.

## 6 Performance Overview

We position our results relative to existing prior work in Table 7. We extracted the results from the referenced papers, with the exception of the Fuyu-8B (Bavishi et al., 2023) model. We performed our own evaluation as the authors have not provided the results on the ChartQA benchmark.

Our work significantly outperforms prior models specialized on the ChartQA benchmark. Concurrent to our work, ChartLlama-13B also uses synthetic data generated, but with a fairly different approach. Although outside the scope of our work, it may be that the approach took to train the much smaller MatCha and UniChart models may be combinable with the approach we presented in this work, leading to possible improved performance with even less computational resources.

The method introduced in this work can be uniquely combined with much larger models through rationale generation. As shown in the results, rationales generated by VLMs can suffice for larger LLMs to effectively operate on, providing a text-representation of the chart conditioned on the question. Our method matches the recently introduced Gemini Ultra model and outperforms previous approaches.

## 7 Future Work

We highlighted several drawbacks of our approach in Section 5.4. The training mixtures do not have examples where colors are used to construct reasoning examples. Bootstrapping such examples, for example by running a smaller sized model with questions that extract color related information, then combines them, would likely improve quality. Very complex reasoning examples are also limited. Specifically, semantically identifying chart elements and performing numeric computations to solve questions would further improve quality.

## 8 Conclusion

We introduced a novel recipe that significantly improves the reasoning capabilities of VLMs. Applied to PaLI-3, our method significantly outperforms even the 10x larger PaLI-X on the ChartQA benchmark, establishing a new state-of-the-art. We demonstrate how the pre-training stage improves downstream performance. Our synthetic data generation technique coupled with the use of a multi-task setup, successfully transfers reasoning capabilities from larger LLMs to smaller VLMs. Moreover, our method enables a computationally more expensive setup where predicted rationales are refined using program-of-thoughts with PaLM 2-S. The composite solution outperforms Gemini Ultra and GPT-4V on the ChartQA benchmark.

## 9 Limitations

We acknowledge limitations of our approach.

**Table representation** Although our final model works on pixels only, our synthetic data generation method requires having access to a table version of the charts for leveraging LLMs to construct rationales, additional question/answer pairs, etc for the training datasets. Although it is likely that inferred tables or output of an OCR model may replace to some degree the presence of gold tables, it will likely affect final model quality.

**PaLI-3** The pre-training and fine-tuning recipe for synthetic data creation, as well as the training methodology should be applicable broadly on open source models as well. However, we acknowledge that the choice of PaLI-3, a proprietary flavor of VLMs, is not as a good of a choice as an open source flavor available externally.

**Risks associated with synthetic dataset** Since the method for constructing our dataset relies on LLMs, there are certain inherent risks that come with that, for example that of hallucination. Although our technique extends the publicly available ChartQA dataset, additional care needs to be taken into account when planning to apply it for releasing models or dataset openly. Although the metrics are state-of-the-art, it cannot be guaranteed that model outputs can't be abused if trained in this manner.

**Reasoning limitations** We acknowledge limitations stemming from the empirical prompt creation process, which is based on human inspection of model errors. LLM capabilities used for the synthetic data creation, although impressive, continue to have numerous limitations as reported by the community.

## 10 Acknowledgements

We thank Srinivas Sunkara and Maria Wang for their contributions on the infrastructure that enabled us to run these experiments. Further, we thank Xi Chen for his tireless support and insights into PaLI-3 details and training recipes and Cheng-Yu Hsieh and Yasuhisa Fujii for the detailed discussions on the multi-task setup. Daniel Keysers and Radu Soricut have provided detailed feedbacks that significantly improved the paper. Matt Sharifi and Ewa Dominowska provided senior leadership support for this work.

Lastly, feedback from anonymous reviewers rPKR and 453J motivated running additional experiments further strengthening the contribution of this work by showcasing the method is generally applicable.

## References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, and Eliza Rutherford et al. 2022. Flamingo: a visual language model for few-shot learning.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, and Laurent El Shafey et al. 2023. PaLM 2 Technical Report.

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. 2023. Introducing our multimodal models.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023a. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks.

Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, and Siamak Shakeri et al. 2023b. Pali-x: On scaling up a multilingual vision and language model.

Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, and Piotr Padlewski et al. 2023c. Pali-3 vision language models: Smaller, faster, stronger.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale.

Sebastian Gehrmann, Sebastian Ruder, Vitaly Nikolaev, Jan A. Botha, Michael Chavinda, Ankur Parikh, and Clara Rivera. 2022. Tata: A multilingual table-to-text dataset for african languages.

Gemini Team, Google. 2023. Gemini: A Family of Highly Capable Multimodal Models. https://blog.google/technology/ai/google-gemini-ai/.

Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. 2023. Chartllama: A multimodal llm for chart understanding and generation.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2023a. Deplot: One-shot visual language reasoning by plot-to-table translation.

Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. 2023b. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering.

Junyu Luo, Zekun Li, Jinpeng Wang, and Chin-Yew Lin. 2021. Chartocr: Data extraction from charts images via a deep hybrid framework. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1916–1924.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason.

Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. 2023. Unichart: A universal vision-language pretrained model for chart comprehension and reasoning. *arXiv preprint arXiv:2305.14761*.

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

Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots.

OpenAI. 2023. GPT-4 Technical Report.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023a. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Jialin Wu, Xia Hu, Yaqing Wang, Bo Pang, and Radu Soricut. 2023. Omni-smola: Boosting generalist multimodal models with soft mixture of low-rank experts.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models.

# A Prompts for PaLM-2

We use PaLM 2-S and PaLM 2-L throughout this work. Here we describe the prompts used for the different purposes. Our *ChartQA-Rationale-S* dataset is a straightforward augmentation of the ChartQA dataset, by predicting the rationales using the table, answer and question. For this, we have constructed the prompt illustrated in Figure 4. The *ChartQA-ExtraQAR-S/L* datasets are constructed using PaLM 2-S/L respectively for which we extended the 1-shot prompt provided by (Liu et al., 2023a). We chose this prompt for simplicity and for it already containing several diverse question examples. The prompt is illustrated in Figure 5.

Lastly, we describe an online refinement of the rationale prediction using program-of-thoughts in Section 5.5. For this, we manually constructed the prompt illustrated in Figure 6. This was built by inspecting a few validation errors when the numeric values computed by the VLM were wrong.

# B Generated Examples

**Licensing** As we redistribute certain data artifacts, we note that the ChartQA dataset at the time of this writing is marked as GPL v3.0 [3]. In this section we provide visual examples of our synthetically generated training datasets, using PaLM 2-S/L models, as well as the programmatically generated templates for mathematical computations. Figure 7 contains an example of synthesized example using only the table representation. The question, answer and rationale cover an aspect of the table and are generated together with 3-5 other questions.

Figure 8 and Figure 9 are examples of a programmatically generated questions based on the template to compute the mean. The markdown table provided as input is processed through a function that takes the corresponding values and outputs all the elements, including the reasoning trace in the rationale for computing the mean as shown in the figure.

# C Model Outputs

In this section we provide examples that accompany our analysis of the model behavior. We highlighted impressive performance, as can be seen in Figure 11, Figure 12 and Figure 13. However, we noted several limitations as well, as can be seen in Figure 10, Figure 14 and Figure 16.

[3]https://github.com/vis-nlp/ChartQA/blob/main/LICENSE

```
You are given a table, a question and answer and your task is to output a
rationale that justifies why the answer to the question is the one provided.

Question: What was the unemployment rate in Poland in 2020?
Table: TITLE |
Characteristic | Unemployment rate
2020 | 3.04%
2019 | 3.47%
2018 | 3.85%
2017 | 4.89%
2016 | 6.16%
2015 | 7.5%
2014 | 8.99%
2013 | 10.33%
2012 | 10.09%
2011 | 9.63%
2010 | 9.64%
2009 | 8.17%
2008 | 7.12%
2007 | 9.6%
2006 | 13.84%
2005 | 17.75%
2004 | 19.07%
2003 | 19.37%
2002 | 19.9%
2001 | 18.37%
2000 | 16.31%
1999 | 12.29%
Answer: 3.04
Rationale: The table is about the unemployment rate in Poland from 2020 to
1999. The unemployment rate in Poland in 2020 is 3.04%.

Question: Is the difference in import value between fiscal year 2020 and
fiscal 2018 larger than the difference between 2013 and 2011?
Table: TITLE |
Characteristic | Import value in billion Indian rupees
FY 2020 | 1590.66
FY 2019 | 1888.81
FY 2018 | 2209.7
FY 2017 | 1594.6
FY 2016 | 1314.1
FY 2015 | 1379.68
FY 2014 | 1442.93
FY 2013 | 1231.68
FY 2012 | 1343.74
FY 2011 | 1541.37
Answer: No
Rationale: The difference in important value between 2020 and 2018 is
1590.66-2209.7 = -619.04. Between 2013 and 2011 is 1231.68-1541.37 =
-309.69. Because -619.04 is smaller than -309.69 the answer is no.

Question: What was the revenue from sponsorship, licensing and merchandising
at the 2008 EURO in Switzerland and Austria?
Table: TITLE |
Characteristic | Revenue in million euros
2016 France | 483.3
2012 Poland & Ukraine | 313.9
2008 Switzerland & Austria | 289.8
2004 Portugal | 182.2
2000 Belgium & the Netherlands | 54.1
1996 England | 29.3
1992 Sweden | 9.7
Answer: 289.8
Rationale: From the table, the revenue from sponsorship, licensing and
merchandising at the 2008 EURO in Switzerland and Austria is 289.8.

Question: How many people in Sub-Saharan Africa had no access to electricity
in 2016?
Table: TITLE |
Characteristic | 2009 | 2016 | 2030
Central and South America | 30 | 17 | 4
North Africa | 1 | 0 | 0
Sub-Saharan Africa | 586 | 588 | 602
Middle East | 21 | 17 | 14
India | 289 | 239 | 0
China | 8 | 0 | 0
Rest of developing Asia | 329 | 200 | 54
Answer: 588
Rationale: The table is about the number of people in different regions who
had no access to electricity in 2009, 2016 and 2030. For Sub-Saharan Africa
the values are 586 for 2009, 588 for 2016 and 602 for 2030. Therefore the
answer is 588.

Question: {question}
Table: {table}
Answer: {answer}
Rationale:
```

Fig. 4: The input template, with a 4-shot prompt, for generating the ChartQA-Rationale-S dataset using PaLM 2-S.

```
You are a helpful assistant who creates unique and innovative
question-answer pairs for training other models.

You should create question-answer pairs from text tables. The questions can
be of two types: (i) directly answered from the table, and (ii) inferred by
applying simple mathematical operations on the values in the table. The
mathematical operations can include maximum, minimum, average, peak, etc.

The questions may not always be answerable from the give table. If a
question can be answered, the answer itself should be crisp and unambiguous.
The answer should be preceded by a brief description detailing how the
answer was arrived at. So the answer format is:
Rationale: ...
Answer: ...

An example table and some sample QA pairs are shown below.

Favor rates of US political parties
Year | Democrats | Republicans | Independents
2004 | 68% | 45% | 53%
2006 | 58% | 42% | 53%
2007 | 59% | 38% | 45%
2009 | 72% | 49% | 60%
2011 | 71% | 51% | 58%
2012 | 70% | 48% | 53%
2013 | 72% | 41% | 60%

Q: In which year republicans have the lowest favor rate?
Rationale: Let's find the column of republicans. Then let's extract the
favor rates, they [45, 42, 38, 49, 51, 48, 41]. The smallest number is 38,
that's row 3.  Row 3 is year 2007.
Answer: 2007

Q: What is the sum of Democrats' favor rates of 2004, 2012, and 2013?
Rationale: Let's find the rows of years 2004, 2012, and 2013. We find Row 1,
6, 7. The favor dates of Demoncrats on that 3 rows are 68, 70, and 72.
68+70+72=210.
Answer: 210

Q: By how many points do Independents surpass Republicans in the year of
2011
Rationale: Let's find the row with year = 2011. We find Row 5. We extract
Independents and Republicans' numbers. They are 58 and 51. 58-51=7.
Answer: 7

Q: Which group has the overall worst performance?
Rationale: Let's sample a couple of years. In Row 1, year 2004, we find
Republicans (column 3) having the lowest favor rate 45 (45<68, 45<53). In
year 2006, Row 2, we find Republicans (column 3) having the lowest favor
rate 42 (42<58, 42<53). The trend continues to other years.
Answer: Republicans

Q: Which party has the second highest favor rates in 2007?
Rationale: Let's find the row of year 2007, that's Row 3. Let's extract the
numbers on Row 3: [59, 38, 45]. 45 is the second highest. 45 is the number
of Independents.
Answer: Independents

Q: What was the favor rates for democrats in 2008?
Rationale: Let's find the row of year 2008. Because 2008 is not in the
table, the answer is not known from this data
Answer: None

Q: What is the value of the brown line?
Rationale: Because I don't have color information on the table, the answer
is not known from this data
Answer: None

Depending on the size of the table, you should create 3-7 such QA pairs.
Make sure that you output only the QA pairs and nothing else.

Now create QA pairs for the following table:

{table}
```

Fig. 5: The input template, with a 1-shot prompt, for generating the ChartQA-ExtraQAR-S/L datasets using PaLM 2-S/L.

```
You are a helpful assistant which helps extract the equations from a text
and write python code to fix the result that is usually incorrect in the
text.

You only output valid python code and nothing else. If there is no
arithmetic computation or equation in the solution, you output 'skipped'.

Question: What is the average number of users across properties?
Solution: Facebook has 563 users, Whatsapp has 69 and Instagram 23. The
average number of users across properties is (563 + 69 + 23) / 3 = 2.
Code:
facebook_users=563
whatspps_users=69
instagram_users=23
result['value'] = (facebook_users + whatsapp_users + instagram_users) / 3

Question: What is the average percentage of people using Google in 2015 and
Bing in 2017 at 60+?
Solution: The percentage of people using Google at 60+ in 2015 is 23% and
the percentage of people using Bing in 2017 at 60+ is 8%. 23% + 8% = 42%.
Code:
google_percentage_2015=23
bing_percentage_2017=8
result['value'] = (google_percentage + bing_percentage) / 2

Question: What is WarnerMedia's global revenue for advertising from 2018 to
2019?
Solution: The table is about WarnerMedia's global revenue for advertising,
subscription and content and other from 2018 to 2019. The sum of all the
WarnerMedia's global revenue for advertising from 2018 to 2019 is 2266 +
7408 + 839 = 10299.
Code:
advertising_revenue=2266
subscription_revenue=7408
content_and_other_revenue=839
result['value'] = advertising_revenue + subscription_revenue +
content_and_other_revenue

Question: What is the ratio between oppose and favor distribution for a ban
on high-capacity ammunition clips?
Solution: The table shows the percentage of people who favor or oppose each
of the following gun measures. The percentage of people who oppose a ban on
high-capacity ammunition clips is 47. The percentage of people who favor a
ban on high-capacity ammunition clips is 50. The ratio between oppose and
favor distribution for a ban on high-capacity ammunition clips is 47 / 50 =
0.092361111
Code:
oppose_percentage=47
favor_percentage=50
result['value'] = oppose_percentage / favor_percentage

Question: {question}
Solution: {text}
Code:
```

Fig. 6: The input template, with a 4-shot prompt, for refining arithmetic operations with python code using PaLM 2-S.

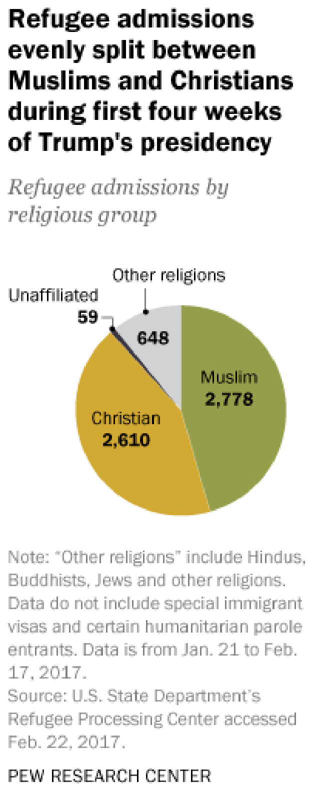
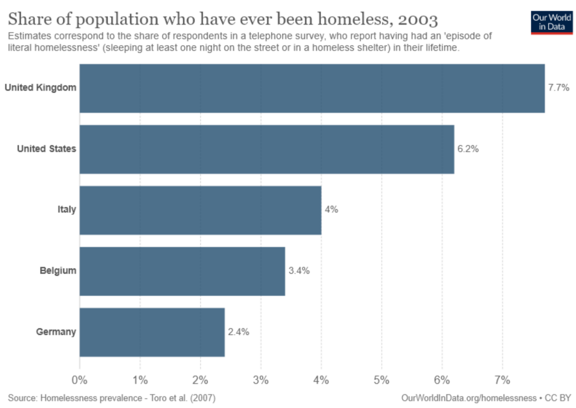



**Table**: "TITLE | Refugee admissions evenly split between Muslims and Christians during first four weeks of Trump's presidency Refugee admissions by religious group \nEntity | Value\nChristian , | 4118\nMuslim , | 2778\nUnaffiliated | 59\nOther religions | 648"

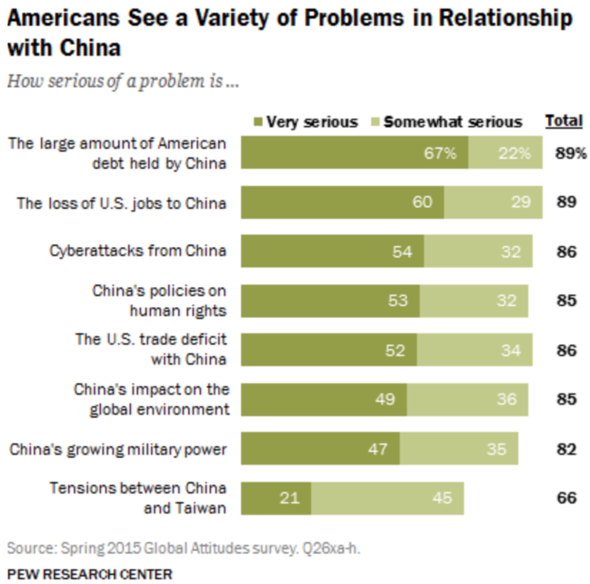
**Question:** "How many more Muslims were admitted than Other religions?"
**Rationale**: "To answer this question, we need to find the number of Muslims and Other religions. The number of Muslims is 2778. The number of Other religions is 648. 2778-648=2130."
**Answer:** 2130

Fig. 7: *ChartQA-ExtraQAR-S/L*: Example of synthesized (Question, Answer, Rationale) pair with PaLM-2 using the table



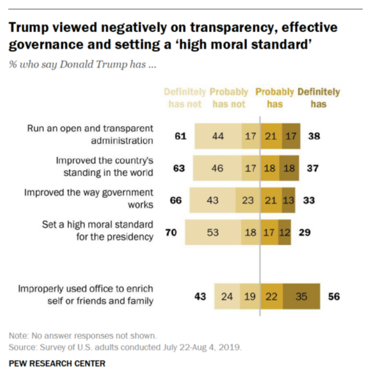

**Question**: What is the average "Share of population who have ever been homeless, 2003 amount?"
**Rationale**: "To answer What is the average Share of population who have ever been homeless, 2003 amount? First extract all the values of Share of population who have ever been homeless, 2003. The values are [7.7, 6.2, 4.0, 3.4, 2.4]. (7.7 + 6.2 + 4.0 + 3.4 + 2.4) / 5 = 4.73. Hence the mean of Share of the population who have ever been homeless, 2003 is 4.73.
**Answer:** 4.73

Fig. 8: *ChartQA-ArithmQAR*: Example of programmatically generated (Question, Answer, Rationale) pair



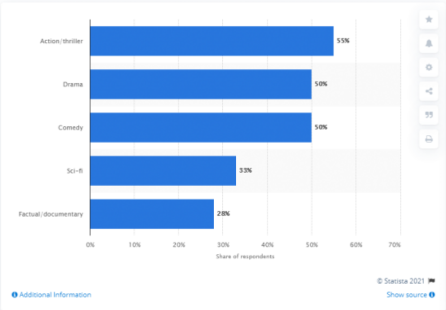

**Question:** What is the average of "Somewhat serious" and "Very serious" for entity "The loss of U.S jobs to China"?
**Rationale**: The value of "Somewhat serious" for entity "The loss of U.S. jobs to China" is 29. The value of "Very serious" for Entity "The loss of U.S jobs to China" is 60. (29 + 60) / 2 = 44.5. The answer is 44.5"
**Answer:** 44.5

Fig. 9: *ChartQA-ArithmQAR*: Example of programmatically generated (Question, Answer, Rationale) pair



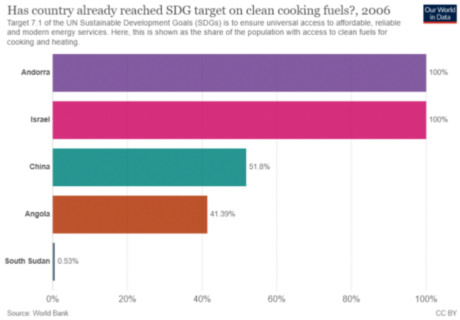

**Question:** What's the maximum value in the brightest yellow bar?
**Answer** [baseline]: **56**
**Answer** [ours]: **22**
**Rationale** [ours]: The **brightest yellow bar is the one for "Improved the country's standing in the world"**. The maximum value in this bar is 18.

Fig. 10: Matching the colors with content is weak.



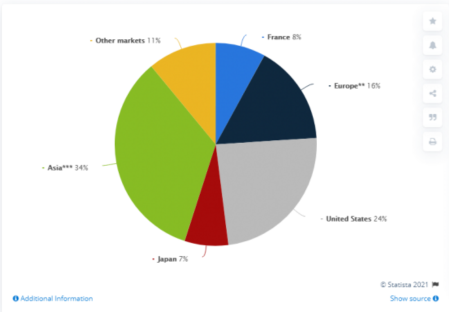

**Question:** What is the difference between the most popular and least popular film genres in the United Kingdom (UK) as of October 2013?
**Answer** [baseline]: 55
**Answer** [ours]: 27
**Rationale** [ours]: The table shows the share of respondents who watched each film genre in the UK as of October 2013. The most popular film genre is Action/thriller, with a share of 55%. The least popular film genre is factual/documentary, with a share of 28%. The difference between the two is 27%.

Fig. 11: Excellent extraction of intermediate values.



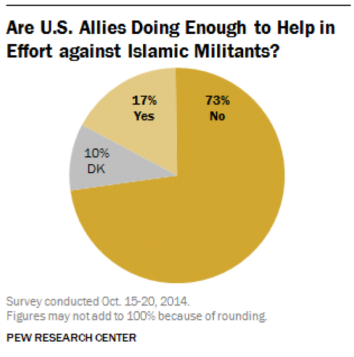

**Question:** How many bars with the equal value in the bar graph?
**Answer** [baseline]: 3
**Answer** [ours]: 4
**Rationale** [ours]: The table shows the percentage of countries that have already reached a certain percentage of their GNI target on clean cooking fuels in 2006. There are four countries with the equal value of GNI target on clean cooking fuels: Israel, China, Angola and South Sudan.

Fig. 14: Checking for equality among multiple values requires even better reasoning abilities.



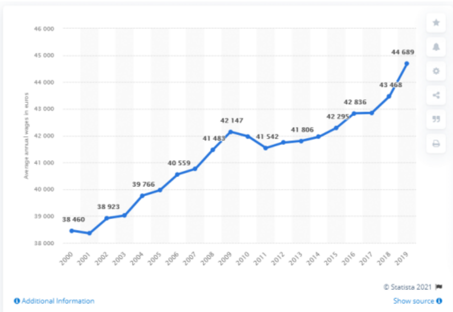

**Question:** What is the total percentage revenue share of the LVMH Group worldwide in 2020 from Asia and Japan?
**Answer** [baseline]: 46
**Answer** [ours]: 43
**Rationale** [ours]: The table shows the revenue share of the LVMH Group worldwide in 2020 from different regions. The total percentage revenue share of the LVMH Group worldwide in 2020 from Asia and Japan is 34% + 7% = 43%

Fig. 12: Correct handling of extracted quantities.



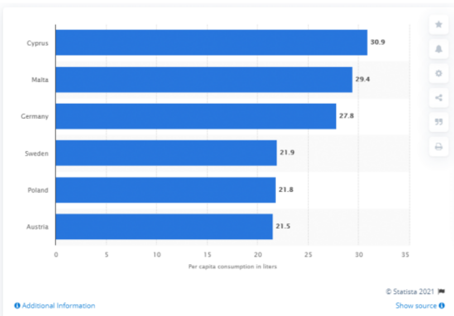

**Question:** IS the value of No more then sum of Yes and Dk?
**Answer** [baseline]: No
**Answer** [ours]: Yes
**Rationale** [ours]: The value of No is 73. The sum of Yes and DK is 17 + 10 = 31. Therefore, the value of No is more than the sum of Yes and DK.

Fig. 15: Despite incorrect arithmetic results, final answer can still be correct




**Question:** What is the difference between maximum and minimum average annual wage over the years?
**Answer** [baseline]: 42421
**Answer** [ours]: 18267 *[inexact, but correct because it's within 5% of ground truth]*
**Rationale** [ours]: The maximum average annual wage is 44690 euros in 2019 and the minimum average annual wage is 38370 euros in 2001. So the difference is 44690 - 38370 = 16815

Fig. 13: Strong ability to infer missing values.




**Question:** Which country has 0.3 difference when compared with Poland?
**Answer** [baseline]: austria
**Answer** [ours]: malta
**Rationale** [ours]: The table shows the per capita consumption in liters in different countries. Poland has a per capita consumption of 21.8 liters. Cyprus has a per capita consumption of 30.9 liters. The difference between the two is 0.3

Fig. 16: Both answer and rationale can be wrong when it comes to enumerating values and checking more complex numerical conditions.

## D Synthetic Dataset: Statistics

We report the final dataset distribution for our synthetically generated examples. We follow the type of descriptions reported in the paper introducing the benchmark (Masry et al., 2022). In Table 2 we describe the type of question generated. We note that we cannot generate, due to the use of a text-based language model, Visual and Compositional and Visual questions, so we only have Data Retrieval and Compositional. The use of visual captions may enable generating these other types.

We report in Table 11 and Table 12 the number of examples (question, answer, rationale) generated for each type of chart and source. We note that the total number of chart images does not change and they are the original ones from ChartQA.

| Source/Graph Type | Pew | Statista-Hum | OWID | OECD | Statista-Aug | Total |
|---|---|---|---|---|---|---|
| H-Bar | 2390 | 1140 | 1270 | - | 13811 | 18611 |
| V-Bar | - | 4999 | - | 358 | 31101 | 36458 |
| Pie | 1162 | 1416 | 4 | - | 366 | 2948 |
| Line | 1101 | 1615 | 663 | 270 | 5190 | 8839 |

Table 11: Frequency of examples by chart types and sources for **ChartQA-ExtraQAR-L**.

| Source/Graph Type | Pew | Statista-Hum | OWID | OECD | Statista-Aug | Total |
|---|---|---|---|---|---|---|
| H-Bar | 3561 | 1594 | 1923 | - | 18777 | 25855 |
| V-Bar | - | 7061 | - | - | 42776 | 49837 |
| Pie | 1525 | 1866 | 6 | 504 | 366 | 4375 |
| Line | 1468 | 2268 | 1075 | 410 | 7406 | 12627 |

Table 12: Frequency of examples by chart types and sources for **ChartQA-ExtraQAR-S**.