Multimodal Chart Retrieval: A Comparison of Text, Table and Image Based Approaches

Anonymous ACL submission

Abstract

We investigate the task of multimodal chart retrieval. Starting from the assumption that images of charts are a visual representation of an underlying table, we propose TAB-GTR, a text retrieval model with table structure embeddings, which achieves state-of-the-art results on NQ-TABLES, improving R@1 by 4.4 absolute points. We then compare three approaches for query to chart retrieval: (a) an OCR pipeline followed by TAB-GTR text retrieval; (b) a chart derendering model, DEPLOT, followed by TAB-GTR table retrieval; (c) a direct image understanding approach, based on PALI-3, a vision language model. We find that the DEPLOT + TAB-GTR pipeline outperforms PALI-3 on in-distribution data, and is significantly more efficient, with 300M trainable parameters compared to 3B of the PALI-3 encoder. However, the setup fails to generalize to out-of-distribution regimes. We conclude that there is significant room for improvement in the chart derendering space, in particular in: (a) chart data diversity (b) richness of the text/table representation.

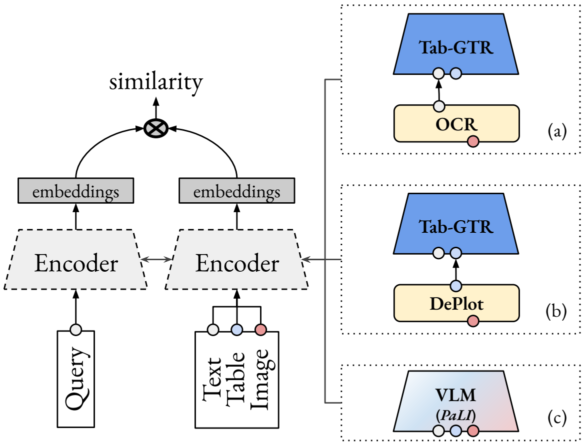


Figure 1: A graphical overview of the three text to chart retrieval approaches evaluated in this work. We use the same architecture for all the three setups (depicted on the right), that is a symmetric bi-encoder dual tower setup, with weights sharing. We train the models to optimize in-batch contrastive loss, without using hard negatives, and evaluate three different approaches: (a) OCR→Text Retrieval; (b) Chart DeRendering→Table Retrieval; (c) VLM Retrieval. The components in yellow are used as black-box modules, and are responsible of converting the image modality (e.g. red circle) to text (a) or table format (b), (e.g. respectively white and azureish circle). This is not needed for (c), as the model can directly handle all the three modalities.

1 Introduction

Multimodal retrieval is the task of retrieving a relevant piece of information from a multimodal dataset, given a query. This task has been extensively studied in the context of text and image retrieval (Yu et al., 2022) or text and table retrieval (Herzig et al., 2021; Kostić et al., 2021), but has received relatively little attention in the context of visually grounded images such as charts and scientific figures. Charts are an important source of information in scientific and technical domains. They are often used to summarize complex data, communicate insights (Hsu et al., 2021; Obeid and Hoque, 2020) as well as for interpreting complex domains such as finance data-analysis, news reporting, and scientific domains (Siegel et al., 2016). However, finding relevant information can be challenging, especially when the query is not specific or decontextualized (Choi et al., 2021).

To the best of our knowledge, this work is the first to investigate multimodal retrieval on chart images, addressing the limited research in this domain. We begin by establishing a powerful table retrieval model that serves as a backbone for subsequent experiments, starting from the assumption that images of charts are a visual representation of an underlying table. To this end, we propose extending text retrieval models with row and column embeddings modeling the table structure, borrowing the main ideas from Herzig et al. (2020);

Andrejczuk et al. (2022). Our proposed model, TAB-GTR, achieves state-of-the-art results on the NQ-TABLES dataset (Herzig et al., 2021), resulting in an improvement of 4.4 absolute points in R@1. For chart retrieval, we compare three approaches, leveraging existing findings in the literature, as also graphically summarized in Figure 1:

(a) **OCR→Text Retrieval**. An OCR model, namely Tesseract (Smith, 2007), converts the chart image into a textual representation. The text is then processed by a text retrieval model, that is TAB-GTR.
(b) **Chart DeRendering→Table Retrieval**. A chart de-rendering model, namely DEPLOT (Liu et al., 2023a), converts the chart image into a table representation. The table is then processed by a table retrieval model, that is TAB-GTR.
(c) **VLM Retrieval.** A vision language model (VLM), such as PALI-3 (Chen et al., 2023) is used for chart retrieval, directly leveraging the content of the chart image.

We evaluate the three approaches on a dataset of charts. Due to the lack of available chart retrieval data, we adapt the CHARTQA, SCICAP, and CHART2TEXT datasets for retrieval. Our extensive experimentation shows that a chart derendering pipeline coupled with a table retrieval model outperforms the VLM setup, when applied in-distribution data (e.g. CHARTQA). However, DEPLOT fails to generalize to more complicated charts (e.g. SCICAP), where it falls behind an OCR baseline.

We conclude analyzing the shortcomings of the chart derendering model suggesting that future work in this area should focus on developing more robust chart derendering pipelines that are able to handle a wider range of chart types and annotations. If realized these improvements can enable (a) more efficient resource utilization, as DEPLOT + TAB-GTR pipeline is significantly more efficient, with $300M$ trainable parameters compared to $3B$ of the PALI-3 encoder; (b) flexible applications of 0-shot chart derendering with large language prompting/retriever models, as done in (Liu et al., 2023b).

2 Related work

Text / Table Retrieval. Text retrieval has been extensively studied in the literature (Karpukhin et al., 2020; Ni et al., 2022). In this work, we build upon existing work and repurpose a generalizable text retriever model to work on table inputs, following the same ideas of Herzig et al. (2020) and Andrejczuk et al. (2022). By building on top of a pre-trained text retrieval model (Ni et al., 2022) we achieve better performance than (Herzig et al., 2021) and (Kostić et al., 2021), without the need for hard-negative mining or more complex tri-encoder setup. Although the task of table retrieval is not new (Liu et al., 2007), to the best of our knowledge, there is no method that adapts the methodology for the task of chart retrieval.

Chart Retrieval. Existing academic chart retrieval approaches only use metadata about figures, such as the caption text or mentions in the body text, to respond to queries (Xu et al., 2008; Choudhury et al., 2013; Li et al., 2013). Other more recent works, focus on chart to chart retrieval. Xiao et al. (2023) propose a user intent-aware framework for retrieving charts that considers both explicit visual attributes and implicit user intents. However, in this scheme the query is a chart rather than a textual query, limiting the usefulness of the task. Similarly, Ye et al. (2022) use neural image embedding to facilitate exploration and retrieval of visualization collections based on visual appearance. To the best of our knowledge, our work is the first to investigate text query to chart retrieval, focusing on understanding the content of figures.

3 Problem setup

We consider multimodal retrieval problems where a textual query is used to retrieve a document that can be a table, an image (specifically of a chart) or a combination of both.

3.1 Datasets

Due to lack of table and chart retrieval datasets we re-purpose datasets meant for question answering (QA), captioning or summarization. We use the following datasets, whereas general dataset statistics are summarized in Table 1.

NQ-TABLES (Herzig et al., 2021) A table question answering dataset created by filtering Natural Questions (Kwiatkowski et al., 2019) to only include questions for which the answer is contained in a table.

CHARTQA (Masry et al., 2022) A chart question answering dataset with charts gathered from Statista (statista.com), Pew (pewresearch.org),

Dataset	Table data	Image data	Type	Train examples	Eval queries	Eval candidates
NQ-TABLES	✓	×	QA	9594	1068	169 898
CHARTQA (human)	✓	✓	QA	7398	1228	625
CHARTQA (augmented)	✓	✓	QA	20 901	1235	987
CHART2TEXT (Statista)	✓	✓	Summarization	24 368	5222	5222
CHART2TEXT (Pew)	×	✓	Summarization	6500	1393	1393
SCICAP	×	✓	Captioning	333 442	41 410	41 682

Table 1: Datasets used in the paper. NQ-TABLES is used for assessing the quality of TAB-GTR, whereas the other datasets are used for benchmarking chart retrieval.

OWID (ourworldindata.org) and OECD (oecd.org). This dataset has two splits: "human" with human-written question-answer pairs and "augmented" with generated question-answer pairs.

CHART2TEXT (Obeid and Hoque, 2020) A chart summarization dataset of charts extracted from Statista and Pew with human-annotated textual summaries of the chart.

SCICAP (Hsu et al., 2021) A chart captioning dataset consisting of figures and figure captions extracted from scientific papers.

Some datasets (CHARTQA and the Statista subset of CHART2TEXT) include human-annotated **gold tables** representing the data on the chart. For each dataset we use the text (i.e. question, transcript or caption) as the **query** and the image plus when available the table as the retrieval **candidate**.

For training we treat each original training set example as a positive query-candidate pair. For evaluation we need a set of queries, a set of candidates and an assignment of the gold candidate to each query. For all datasets we use the evaluation set (dev or test) as the source of queries and gold candidates. Queries and candidates are deduplicated by exact match.

On NQ-TABLES we use all tables (train, dev and test) as evaluation candidates, following (Herzig et al., 2021). These tables are deduplicated by string similarity as in (Herzig et al., 2021).

3.2 Evaluation

We use standard retrieval metrics, reporting recall at k (R@k), mean average precision (MAP) and the highest F1 score over any classification threshold (picked separately for each dataset). We report single run numbers as we have not seen significant variance between runs. We report the final numbers on the test sets, with the exception of NQ-TABLES for which we report dev set numbers in accordance with previous literature. We have used the dev sets for development and model selection.

3.3 Contextual queries.

QA datasets may include contextual queries, that is, queries formulated in the context of the chart. These queries are highly ambiguous and including them in the dataset adds noise to the training and evaluation metrics. To overcome this issue in a text passage setup, Choi et al. (2021) propose the use of decontextualizer model. To evaluate the scope of the problem and feasibility of this solution we have we have manually classified 50 examples from each split of CHARTQA into one of a few categories:

1. **Not contextual**, e.g. *"How many people from the age group 80 years and above have died due to COVID in Italy as of June 8, 2021?"*.
2. **Decontextualisable from text**, e.g. *"When does the gap between the two countries reach the smallest?"*. These can be decontextualized based on the text appearing on the chart and deplotted table data.
3. **Decontextualisable visually**, e.g. *"What's the peak value of dark brown graph?"*. These can be decontextualized but require additional visual information from the chart, i.e. colors.
4. **Missing context**, e.g. *"What is the ratio of yes to no?"* with a chart that does not include specific labels for the *"Yes"/"No"* categories.
5. **Inherently contextual**, which include queries that ask for specific visual or mathematical reasoning on the chart and cannot be decontextualized, e.g. *"What category does the red color indicate?"* or *"Are there any two bars having the same value?"*.

The results in Table 2 show that in CHARTQA (human) 70% of queries are contextual and text-only decontextualisation would only partially address this problem, leaving out 42% of all queries. Given the lack of a comprehensive solution, and to avoid further complexity, we have kept the data as-is.

We have not found this to be a problem in the other datasets: the CHARTQA (augmented) split is mostly non-contextual. NQ-TABLES queries are Google search queries from Natural Questions (Kwiatkowski et al., 2019), stated without context. Captions and summaries are highly informative about the content of the chart and do not present the same ambiguity problems.

	CHARTQA (h)	CHARTQA (a)
Not contextual	30%	94%
Decontextualisable from text	28%	0%
Decontextualisable visually	12%	0%
Missing context	4%	0%
Inherently contextual	26%	6%

Table 2: Analysis of query contextuality on CHARTQA. We have manually labeled 50 examples from each dataset. The augmented split queries are mostly non-contextual. In the human split 30% are non-contextual, 40% could be decontextualised based on textual or visual information from the chart and 30% cannot be decontextualised or are missing necessary context.

Other datasets. We decided against using PlotQA (Methani et al., 2020) because of its synthetic/template-based nature and focus on reasoning over a specific chart and high percentage of contextual queries (estimated by us to be around 70%). However the data might still be useful after filtering and decontextualisation, or as noisy chart retrieval pre-training data.

4 Table Retrieval with TAB-GTR

We present TAB-GTR, a multimodal extension of the GTR (Ni et al., 2022) model that handles both text and tabular data. We extend the T5 encoder architecture following the approach of (Herzig et al., 2021; Andrejczuk et al., 2022) by adding two-dimensional positional embeddings that encode the table structure. The overview of the model architecture is shown in Figure 2.

Given an input text t and input table with n columns and m rows and text $c_{i,j}$ in cell at column $1 \leq i \leq n$ and row $1 \leq j \leq m$ we tokenize each piece of text and concatenate them all into a single sequence. For each token we add two additional discrete features **text_col** and **text_row**:

- For tokens in the text t we set both **text_col** $=$ **text_row** $= 0$.
- For tokens in a table cell $c_{i,j}$ we set **text_col** $= i$ and **text_row** $= j$.

Columns and rows are embedded into feature vectors and the embeddings added to the token embeddings before being fed to the transformer encoder. This provides the network with absolute positional embeddings of the table row and column corresponding to the tokens. We also use relative positional attention bias inherited from the T5 architecture, which is based on the linearized token sequence and not aware of the table structure.

4.1 Model details

The only new parameters added to GTR are the column and row embeddings. We set the maximum row and column numbers to be 128, which for the large model results in $2 \times 128 \times 768 \simeq 197$K new parameters, which is negligible compared to the total 334M parameters. We initialize these embeddings from scratch and learn them entirely during fine-tuning on the final task. All the other parameters are initialized from a pre-trained GTR checkpoint. We use a symmetric retrieval model, i.e. the left and right tower share weights. We have not tried an asymmetric setup as the added complexity and memory requirements.

4.2 Evaluation on NQ-TABLES

We evaluate the performance of the TAB-GTR model, as well as vanilla GTR without the extra table structure embeddings, on the dataset NQ-TABLES. We train the models to optimize in-batch contrastive loss, without using hard negatives.

We have tuned the hyperparameters for the GTR model and used the same values for TAB-GTR, as the models are extremely similar. We trained both for 1000 steps with batch size 1024, using the Adafactor optimizer (Shazeer and Stern, 2018) with constant learning rate 0.0003. The dropout rate is set to 0.1 during training.

The evaluation results are in Table 3. The TAB-GTR model achieves state of the art results and significant improvement over GTR, with 89.42% recall at 10 compared to 87.64% of GTR and 86.40% of the best previously published result Kostić et al. (2021).

4.3 Conclusions

The addition of table positional embeddings to a text model achieves a significant improvement at a negligible cost, adding only 0.06% extra model parameters, makes no difference on training times and does not require additional pretraining. According to (Herzig et al., 2020) table positional embeddings

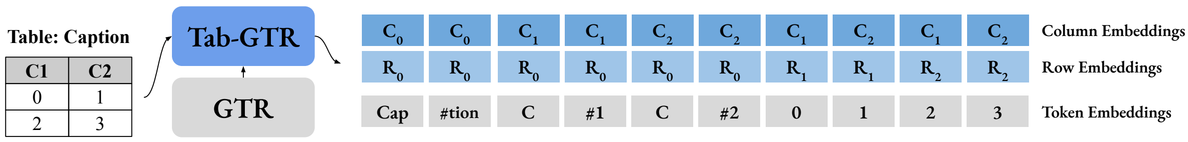


Figure 2: TAB-GTR leverages a GTR checkpoint (Ni et al., 2022) as a backbone model (represented in grey) and adds two dimensional positional embeddings (represented in blue) to represent table structure (i.e. row and columns), as done by Herzig et al. (2020); Andrejczuk et al. (2022). This is a minimal addition in terms of #params, on top of GTR, as the structural embeddings represent $< 0.06\%$ of the total.

Model	NQ-TABLES (dev)			
	R@1	R@10	R@50	R@100
TAPAS, large	35.90	75.90	91.40	N/A
+ hard negatives (Herzig et al., 2020)	44.20	81.80	92.30	N/A
Tri-encoder BERT (Kostić et al., 2021)	N/A	86.40	N/A	96.7
GTR, large (Ni et al., 2022)	44.48	87.64	96.63	97.57
TAB-GTR, large	**48.88**	**89.42**	**97.85**	**98.60**

Table 3: Comparison of table retrieval models on NQ-TABLES (dev split). TAB-GTR is the simplest and best performing model.

also improve performance of models specifically pretrained on table data. That makes this method an obvious inclusion to maximize model performance on table data. Given its strongest performance we will use TAB-GTR as the base text/table model for experiments on chart retrieval.

5 Chart Retrieval

5.1 Models

We compare a direct image understanding approach to approaches using an intermediate text or table representation.

5.1.1 Direct image understanding

For direct image understanding we use PALI-3 (Chen et al., 2023), a $5B$ parameter vision-language model. We discard the decoder and only use the encoder part of the model, consisting of a ViT vision encoder and a text transformer encoder. PALI-3 achieves very strong results on chart understanding tasks such as CHARTQA (Masry et al., 2022), outperforming Matcha (Liu et al., 2023b) and state of the art results on the cross-modal retrieval task XM3600 (Thapliyal et al., 2022).

We use PALI-3 as a symmetric multimodal dual encoder model, keeping both the ViT component and text encoder. We extend the model with table positional embeddings for table inputs (in the same way we did with GTR). Both towers are able to encode text, table and image data. If a modality is not present we simply do not include any tokens corresponding to that modality. Images are padded to a square shape and resized to resolution 448×448 pixels.

5.1.2 Text / Table representation

All text/table-based approaches use TAB-GTR as the base retrieval model. We compare different ways of converting the chart to text or table data.

DEPLOT (Liu et al., 2023a) is a zero-shot image-to-table model trained to recover tabular data underlying a chart. The architecture is based on Pix2Struct (Lee et al., 2023), a ViT model with 282M parameters.

OCR. We use the Tesseract OCR engine (Smith, 2007), which is available as an open source library. We feed model the linearized OCR text output, without any bounding box information.

Gold tables. For comparison we use human-annotated table information present in the CHARTQA and CHART2TEXT (Statista) datasets.

5.2 Training

All models use the AdaFactor optimizer (Shazeer and Stern, 2018) with constant learning rate 0.0003 and bidirectional in-batch softmax cross entropy, as in CLIP (Radford et al., 2021).

Dual encoder training with in-batch negatives is highly sensitive to batch size as the quality of the approximation depends on the sample size. We use the same batch size of 256 for all experiments, as we have found that increasing it further does not give significant improvements.

For each experiment we pick the number of training steps by cross-validation on the dev set: we train the model until the dev set softmax accuracy (i.e. R@1 when viewed form the retrieval angle) stops improving and pick the checkpoint with the highest dev set accuracy.

We start from a **single-task setup** where we train a separate model for each of the tasks. Later we introduce a **multi-task setup** where the model is trained on a mixture of data from all the datasets. Multi-task training poses additional difficulties:

1. The loss depends on the mixture in a complicated way as it changes the distribution of negative samples. In the multi-task setup we consider negative pairs where the query and candidate come from different datasets.
2. The datasets have different sizes and levels of noise and require different early stopping schedule to avoid overfitting.

We propose to design a data mixture by picking sampling weights proportional to the best number of training steps on a given dataset in the single-task setup. To simplify the setup we use only a single set of weights: calculated as the average between the DEPLOT, OCR and PALI-3 models and rounded, shown in Table 4. We note that the weights are roughly proportional to dataset size, except for CHARTQA (human) which overfits quickly and was assigned a lower weight. We think that the overfitting is caused by the high proportion of contextual queries in this dataset. Our mixture design improves robustness to noisy data by lowering their weight in the mixture.

Mixture dataset	Weight	Fraction
CHARTQA (human)	0.75	1.3%
CHARTQA (augmented)	4.0	7%
SCICAP	40.0	69.9%
CHART2TEXT (Statista)	7.5	13.1%
CHART2TEXT (Pew)	5.0	8.7%

Table 4: Mixture weights and fraction of the batch sampled from the given dataset.

6 Experiments

6.1 Chart retrieval approaches

We compare the results of our chart retrieval approaches on the single-task setup in Table 5.

We observe that when gold tables are available TAB-GTR generally outperforms other approaches. We treat this model as an oracle as we are interested in a setting where only the image is available.

DEPLOT + TAB-GTR delivers the strongest results on CHARTQA and CHART2TEXT (Statista). This is confirmed by inspecting the performance of DEPLOT chart to table task in isolation, using Relative Mapping Similarity (RMS) metric proposed in (Liu et al., 2023b). However, the setup is the worst performing for CHART2TEXT (Pew) and SCICAP, as also corroborated by manually inspecting the performance of DEPLOT on a small set of 20 examples. Analysing the errors on these datasets reveals some patterns:

1. Most of Pew charts follow the same format, with a header with title and subtitle and a footer with the data source. This information very often contains distinct keywords that are directly referenced in the summary, which explain the high performance of an OCR approach. This is also in line with the statistics of Table 6. We can clearly see how Pew is the outlier, being the dataset with the highest Query coverage of 0.86.
2. Charts in SCICAP are complex scientific plots, often with multiple subplots. This is a large deviation from the training distribution of DEPLOT which only includes single charts. Some typical error patterns for this dataset can be found in **??**.

PALI-3 underperforms on CHARTQA and CHART2TEXT (Pew). On CHARTQA (human) the performance is below the OCR baseline. We note that dataset in particular is more prone to overfitting and requires more aggressive early stopping; image models generally require more data and so are at a disadvantage here. The low performance on CHART2TEXT (Pew) is surprising given that the pretrained model performs well on OCR tasks.

OCR + TAB-GTR is only competitive in CHART2TEXT (Pew) and ranks 2nd for SCICAP. The former is an outlier according to Table 6, whereas SCICAP seems an out-of-distribution setup for DEPLOT. The Statista charts (in contrast to Pew) contain no title or additional context besides the axis labels and so more reasoning has to be done based on the data represented in the chart. We have also found that the Tesseract OCR can have issues reading small, rotated text.

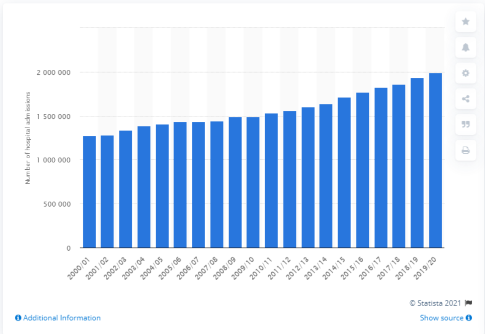
6.2 Multi-task training

We investigate the impact of multi-task training on the model performance, showing the results and difference with respect to the single-task setup in Table 7. We have trained these models on the mixture described in Section 5.2.

Model	ChartQA (human)			ChartQA (augmented)			SciCap			Chart2Text (Statista)			Chart2Text (Pew)		
	R@10	MAP	F1	R@10	MAP	F1	R@10	MAP	F1	R@10	MAP	F1	R@10	MAP	F1
TAB-GTR (gold table)	**64.33**	**52.09**	**52.11**	**97.33**	**82.82**	59.03	N/A	N/A	N/A	**99.10**	**95.45**	**78.48**	N/A	N/A	N/A
TAB-GTR + DePlot	**62.95**	**48.77**	**45.70**	**96.76**	**81.25**	**60.59**	56.55	44.48	46.53	**98.76**	**93.88**	69.04	95.12	82.85	68.81
TAB-GTR + OCR	60.10	45.86	44.57	84.94	63.27	46.88	61.42	48.64	47.55	88.85	68.78	43.44	98.35	**95.84**	**92.01**
PaLI-3	58.88	42.90	37.00	95.14	75.36	49.83	**76.92**	**64.06**	**54.49**	98.12	90.40	**71.32**	**99.35**	92.17	75.59

Table 5: Comparison of the three different approaches to chart retrieval in the single-task setup (last three rows), as graphically depicted in Figure 1. The first row is the oracle setup where the gold table is used instead.

Query: In 2000/01 there were approximately 1.28 million adults admitted to hospital in England due to an illness caused by smoking . By 2019/20 the number of hospital admissions as a result of smoking had increased to approximately 1.99 million , the largest number during the provided time period.



Query: Health care providers at hospitals and medical centers around the country are on the front line of care for those ill with the virus. As Americans take stock of early efforts to control the outbreak, 71% are very or somewhat confident that hospitals and medical centers in their local area can handle patient needs.

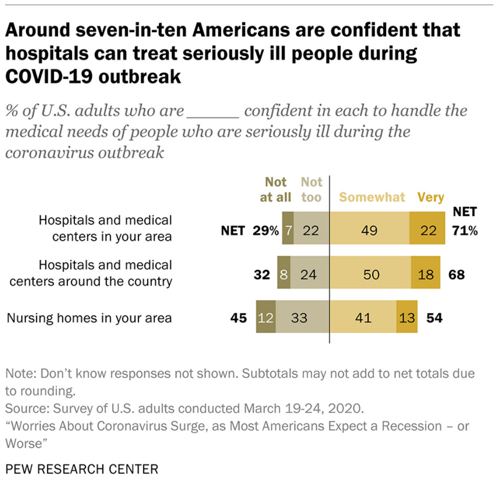


Figure 3: Typical Statista (top) and Pew (bottom) examples from Chart2Text. DePlot performs well on data-heavy examples from Statista but underperforms on text-heavy examples from Pew.

Dataset	Query	OCR	Jaccard index	Query cov.
	(# unique words)			
ChartQA (h)	11	65	.03	.17
ChartQA (a)	12	67	.02	.09
SciCap	31	84	.04	.14
Chart2Text (S)	17	69	.03	.15
Chart2Text (P)	22	80	.26	.86

Table 6: For each dataset we compute the average number of unique words for the Query and text outputted by the OCR model, after a lower case normalization and using whitespace splitting. We report the Jaccard index between Query and OCR, and query coverage defined as percentage of unique words in the query that are covered by the OCR text.

Tab-GTR + DePlot and Tab-GTR + OCR models generally perform worse in the multi-task approach. One possible explanation for this low performance could be the amount of noisy table added to the training mixture, especially for SciCap that has the largest weight in the mixture; another possible cause could be model capacity which is an order of magnitude less for Tab-GTR with respect to PaLI-3.

PaLI-3 shows large benefits on the ChartQA datasets and mostly neutral results on other datasets, with the exception of a large drop in F1 score on the Chart2Text (Statista) dataset. We observe that the drop is caused by a drop in precision as R@1 decreased from 85.48% to 81.90%, while the R@10 remains high.

We note that the ChartQA (human) performance is largely improved for PaLI despite only making up around 3 examples per batch.

6.3 Retrieval with PaLI-3 + DePlot

Given the distinct strengths of the two approaches we consider combining them into a single model. We do so by adding the DePlot tables as an additional input to PaLI-3, encoding them as in Section 4. Results are summarized in Table 8.

The addition is generally an improvement over

Model	ChartQA (human)			ChartQA (augmented)			SciCap			Chart2Text (Statista)			Chart2Text (Pew)		
	R@10	MAP	F1	R@10	MAP	F1	R@10	MAP	F1	R@10	MAP	F1	R@10	MAP	F1
TAB-GTR + DePlot	61.24 (-1.71)	47.75 (-1.02)	**44.43** (-1.27)	**97.73** (+0.97)	**81.42** (+0.17)	**56.00** (-4.59)	57.01 (+0.46)	44.99 (+0.51)	46.42 (-0.11)	**98.66** (-0.10)	**92.25** (-1.63)	**64.05** (-4.99)	93.83 (-1.29)	79.34 (-3.51)	64.43 (-4.38)
TAB-GTR + OCR	59.04 (-1.06)	45.70 (-0.16)	42.78 (-1.79)	86.48 (+1.54)	66.30 (+3.03)	46.77 (-0.11)	61.49 (+0.07)	48.79 (+0.15)	47.89 (+0.34)	87.09 (-1.76)	65.67 (-3.11)	35.63 (-7.81)	98.49 (+0.14)	**95.31** (-0.53)	**86.98** (-5.03)
PaLI-3	**63.93** (+5.05)	**49.01** (+6.11)	42.95 (+5.95)	97.00 (+1.86)	79.32 (+3.96)	54.27 (+4.44)	**77.69** (+0.77)	**63.89** (-0.17)	**54.12** (-0.37)	98.18 (+0.06)	88.08 (-2.32)	59.86 (-11.46)	**99.64** (+0.29)	92.17 (+0.00)	76.53 (+0.94)

Table 7: Results on chart retrieval in the multi-task setup. The numbers in the parentheses show the difference between the multi-task and single-task results.

Model	ChartQA (human)			ChartQA (augmented)			SciCap			Chart2Text (Statista)			Chart2Text (Pew)		
	R@10	MAP	F1	R@10	MAP	F1	R@10	MAP	F1	R@10	MAP	F1	R@10	MAP	F1
TAB-GTR + DePlot (single-task)	**62.95**	**48.77**	**45.70**	96.76	81.25	**60.59**	56.55	44.48	46.53	98.76	**93.88**	69.04	95.12	82.85	68.81
PaLI-3 + DePlot (multi-task)	61.97 (-1.96)	48.52 (-0.49)	45.28 (+2.33)	**97.65** (+0.65)	**82.91** (+3.59)	57.66 (+3.39)	**76.96** (-0.73)	**63.30** (-0.59)	**53.96** (-0.16)	**98.77** (+0.59)	93.13 (+5.05)	**70.71** (+10.85)	**99.78** (+0.14)	**93.15** (+0.98)	**77.94** (+1.41)

Table 8: Results for a PaLI-3 model combining the image input with the DePlot table input. Numbers in parentheses show the difference with respect to a multi-task PaLI-3 model that does not use the deplotted tables. The first model is shown for comparison.

the image-only model, especially on datasets where DePlot performs well according to Table 9. There is a slight consistent decrease in performance on SciCap, which poses the hardest generalization challange for DePlot. The combined model performs well across all tasks, showing both high performance on tasks in DePlot's domain and the capability to better generalize to different chart data (SciCap, Chart2Text (Pew)). Results are generally inline with previous literature research, where adding additional information in addition to image inputs (e.g. OCR text) provide significant improvements (Chen et al., 2022).

Dataset	**Precision**	**Recall**	**F1**
	(DePlot prediction)		
ChartQA (human)	65.24	69.94	67.17
ChartQA (augmented)	89.09	97.59	92.82
Chart2Text (Statista)	87.63	94.55	90.00
Dataset	**Accuracy†** *(manually evaluated)*		
SciCap	15		
Chart2Text (Pew)	35		

Table 9: DePlot performance on the various datasets. For the datasets that provide gold tables as the target, we use the Relative Mapping Similarity (RMS) proposed in (Liu et al., 2023a) to asses the similarity between tables. As gold tables are not available for SciCap and Chart2Text (Pew), we instead report Accuracy† as a proxy metric, manually evaluated on a randomly sampled set of 20 examples.

7 Conclusions

In this paper, we tackle the problem of chart retrieval, which, to the best of our knowledge, has not been explored before, at least in the context of text query to chart retrieval. From the assumption that chart images are visual representations of an underlying table, we establish a SOTA table retrieval backbone, TAB-GTR, combining the findings of Ni et al. (2022); Herzig et al. (2020); Andrejczuk et al. (2022). We then benchmark two table setups for TAB-GTR, with an oracle gold table setup and a table derived by a deplotter model (Liu et al., 2023a). The same deplotted table is also used as inputs along with the chart image through a strong VLM, PaLI-3 (Chen et al., 2023). Our experimentation on 5 datasets shows that if we have access to the underlying table representation, TAB-GTR is the most economical and higher quality option, with a $10\times$ saving in parameter count. With no access to the underlying table the best approach is:

1. Use de-plotting and table retrieval if a high quality deplotter is available. This yields the best results with a small model size.
2. If the data is out of distribution, a VLM delivers better generalization capability, at a much higher computational cost and likely lower performance than if the deplotter was expanded to cover the new domain.

We show that the two approaches provide complementary benefits: a VLM can be extended with deplotter input to achieve both high performance on in-distribution data and better flexibility.

Limitations

The following are the shortcomings of our work, which are presented in a transparent manner to encourage future research.

First, the chart retrieval datasets were not originally created for the retrieval task. Instead, they were adapted for this purpose. Additionally, the chart domains we tested were limited to a few domains (e.g. scientific figures and general statistics). This limitation is inherited from the existing academic chart QA datasets, which only cover a limited number of domains. Therefore, in order to fully assess retrieval performance, it may be beneficial to expand the scope of the work to include other domains (e.g. finance, news, etc.).

Related to the limitation above, we used a deplotter model, specifically DePlot (Liu et al., 2023a), which, as we see in Table 5, does not seem to generalize to other domains. Indeed, OCR baselines, for very out-of-domain datasets, seem to generally perform better. This suggests that future work could focus on improving the robustness of the deplotter module.

Third, we only focused on the English language. We believe that this is an interesting area for future exploration. Datasets such as TaTa (Gehrmann et al., 2023), could be used for follow-up work (unfortunately images are not part of the dataset release).

Despite these limitations, our work represents the first work to explore the problem of chart retrieval. We hope that future research will be able to build upon this foundation.

Ethics Statement

All the data we use is publicly available on the web with appropriate permissive licenses. The chart data has been obtained from publicly available, curated data sources and contains no personally identifiable information (PII) or offensive content. User query data in NQ-Tables has been properly anonymized in (Kwiatkowski et al., 2019). Queries for other datasets have been either written by human annotators or automatically generated and contain no PII or offensive content. The risk is very low as retrieval models have no capability to output novel content, however it might reflect biases present in the datasets.

References

Ewa Andrejczuk, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, and Yasemin Altun. 2022. Table-to-text generation and pre-training with TabT5. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6758–6766, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. 2023. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. 2022. Pali: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

Sagnik Ray Choudhury, Suppawong Tuarob, Prasenjit Mitra, Lior Rokach, Andi Kirk, Silvia Szep, Donald Pellegrino, Sue Jones, and Clyde Lee Giles. 2013. A figure search engine architecture for a chemistry digital library. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 369–370.

Sebastian Gehrmann, Sebastian Ruder, Vitaly Nikolaev, Jan Botha, Michael Chavinda, Ankur Parikh, and Clara Rivera. 2023. TaTA: A multilingual table-to-text dataset for African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1719–1740, Singapore. Association for Computational Linguistics.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. 2021. Open domain question answering over tables via dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 512–519, 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.

Ting-Yao Hsu, C Lee Giles, and Ting-Hao Huang. 2021. SciCap: Generating captions for scientific figures. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3258–3264, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Bogdan Kostić, Julian Risch, and Timo Möller. 2021. Multi-modal retrieval of tables and texts using tri-encoder models. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 82–91, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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.

Zhuo Li, Matthew Stagitis, Sandra Carberry, and Kathleen F McCoy. 2013. Towards retrieving relevant information graphics. In *Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval*, pages 789–792.

Fangyu Liu, Julian 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. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10381–10399, Toronto, Canada. Association for Computational Linguistics.

Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. 2023b. MatCha: Enhancing visual language pretraining with math reasoning and chart derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12756–12770, Toronto, Canada. Association for Computational Linguistics.

Ying Liu, Kun Bai, Prasenjit Mitra, and C Lee Giles. 2007. Tableseer: automatic table metadata extraction and searching in digital libraries. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 91–100.

Ahmed Masry, Xuan Long Do, 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.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Obeid and Enamul Hoque. 2020. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 138–147, Dublin, Ireland. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Noah Siegel, Zachary Horvitz, Roie Levin, Santosh Divvala, and Ali Farhadi. 2016. Figureseer: Parsing result-figures in research papers. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 664–680. Springer.

Ray Smith. 2007. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE.

Ashish V Thapliyal, Jordi Pont Tuset, Xi Chen, and Radu Soricut. 2022. Crossmodal-3600: A massively multilingual multimodal evaluation dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 715–729.

Shishi Xiao, Yihan Hou, Cheng Jin, and Wei Zeng. 2023. Wytiwyr: A user intent-aware framework with multi-modal inputs for visualization retrieval. *arXiv preprint arXiv:2304.06991*.

Songhua Xu, Jamie McCusker, and Michael Krauthammer. 2008. Yale image finder (yif): a new search engine for retrieving biomedical images. *Bioinformatics*, 24(17):1968–1970.

Yilin Ye, Rong Huang, and Wei Zeng. 2022. Visatlas: An image-based exploration and query system for large visualization collections via neural image embedding. *IEEE Transactions on Visualization and Computer Graphics*.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

A Experiment details

All the experiments used bidirectional cross entropy loss with in-batch negatives, batch size of 256, learning rate of 0.0003 and dropout rate of 0.1. Tables 10 and 12 show the number of training steps for each of our models. We stopped training when the validation metric stopped improving. The validation metric used is the average in-batch classification accuracy calculated on the dev set, with batch size 256 and up to 50 batches. For multitask runs we use the average of per-task accuracy weighted by the mixture weights. We show the model size and computational requirements in tables 11 and 13. We estimate that the experiments in this paper cost around $4.9k$ TPU-hours.

Model	Training steps				
	ChartQA (h)	ChartQA (a)	SciCap	Chart2Text (S)	Chart2Text (P)
TAB-GTR (gold table)	200	4500	N/A	4000	N/A
TAB-GTR + DePlot	900	7500	40 000	1500	5000
TAB-GTR + OCR	300	2500	40 000	7000	7000
PaLI-3	1000	2500	40 000	15 000	2000

Table 10: Number of training steps selected by cross validation for single-task training. We stopped SciCap at $40k$ steps because the progress become extremely slow. For model selection we used in-batch accuracy on the dev set.

Model	Batch size	# of TPU chips	TPU-h per 1k steps
Tab-GTR	1024	64	15.20
Tab-GTR	256	16	3.80
PaLI-3	256	64	19.00
PaLI-3 + DePlot	256	128	23.18

Table 11: Model computational requirements. We train our models on the Google Cloud TPU v4. Batch size 1024 is only used for NQ-Tables and all other experiments use batch size 256. All Tab-GTR models (gold, +DePlot, + OCR) use the same sequence length and have the same memory requirements.

Model	Training steps
TAB-GTR + DePlot	76 000
TAB-GTR + OCR	64 000
PaLI-3	68 000
PaLI-3 + DePlot	64 000

Table 12: Number of training steps selected by cross validation for multi-task training. For model selection we used in-batch accuracy on the dev sets aggregated by the mixture weights.

Model	# of weights
DePlot	282M
TAB-GTR	335M
PaLI-3	3 289M

Table 13: Model size. Note that we only use the encoder of PaLI-3 which is why the number of parameters is not $5B$.

B Error examples

In this section we show examples to illustrate the kind of errors the models make. We compare two models side-by-side and show examples where one model returns the correct answer in top k results and the other does not. We use $k = 5$ through this section. A limitation of this method is that it often finds spurious win/loss examples caused by model training stochasticity. To work around that we have manually chosen examples that we think show some error patterns.

B.1 Tab-GTR + DePlot vs Tab-GTR + gold tables

We look at examples where Tab-GTR + DePlot loses to Tab-GTR + gold tables. For this section we only consider datasets with gold tables available. We have found that the two models are very close in performance, however one consistent pattern shown in Figure 4 is that DePlot sometimes omits the title or axis labels.

B.2 PaLI-3 vs Tab-GTR + DePlot

We have found following error patterns for DePlot (shown in Figures 5 to 7):

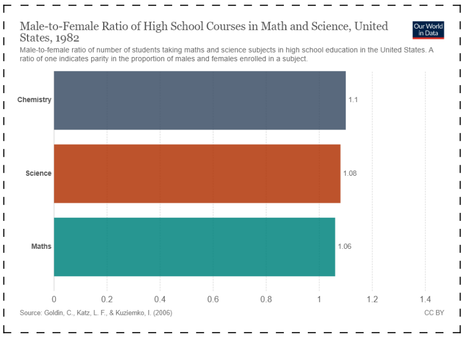
1. Failing to capture text on the chart, such as plot titles or axis labels. This is the same pattern as found in appendix B.1. Examples shown in figs. 5a and 6a.

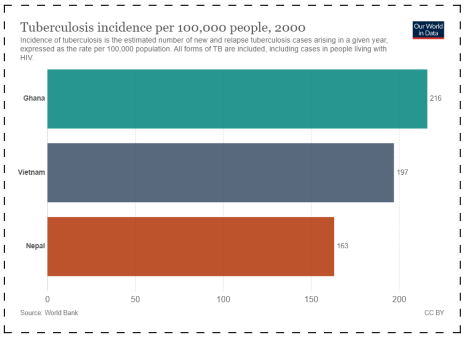
Query

Which subject has the highest male-to-female ratio of High School Courses?

Which place shows the lowest value of Tuberculosis rate?

Chart





Gold Table

Country	Male-to-Female Ratio of High School Courses in Math and Science, United-States, 1982
Chemistry	1.1
Science	1.08
Maths	1.06

Country	Tuberculosis incidence per 100,000 people, 2000
Ghana	216
Vietnam	197
Nepal	163

DePlot Table

Characteristic	Value
Chemistry	1.1
Science	1.08
Maths	1.06

Characteristic	Value
Ghana	216
Vietnam	197
Nepal	163

Figure 4: Select examples from CHARTQA (human) where DEPLOT underperforms with respect to the gold tables. DEPLOT fails to capture the title of the plot.

2. Not capturing visual elements of the chart. On CHARTQA these are usually plot type (e.g. bar, pie) and line colors and we note that these wins are not relevant for retrieval because the queries are highly contextual (fig. 5b). However on SCICAP (figs. 6b, 7a and 7b) PALI-3 is able to recognise more interesting visual information such as semantic content of the chart (e.g. "sigmoid function", "geodesic triangle") or visual placement of the subplots ("left: ..., right: ...").

3. Failing on complex charts with multiple subplots (figs. 6b and 7b). This is a limitation of the training data which only includes single-plot charts.

4. Failures on charts with a very large amount of data points (7b) where DEPLOT tries to capture all individual data points instead of more semantically relevant summary of the chart.

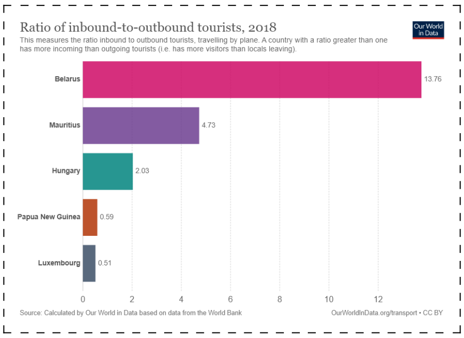
We have not found any specific error patterns for PALI-3. Rather we see that on data that does not trigger the above failure modes TAB-GTR + DEPLOT generally outperforms PALI-3.

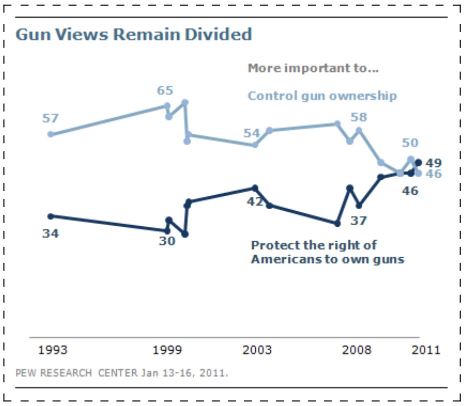
Query

Which place has the highest ratio of tourists ?

Is there a value 30 in the dark blue line?

Chart





Gold Table

Country	Ratio of inbound-to-outbound tourists, 2018
Belarus	13.76
Mauritius	4.73
Hungary	2.03
Papua New Guinea	0.59
Luxembourg	0.51

Year	Control gun ownership	Protect the right of Americans to own guns
1993	0	0
1999	0	0
2003	0	0
2008	58	0
2011	50	49

DePlot Table

Characteristic	Value
Belarus	13.76
Mauritius	4.73
Hungary	2.03
Papua New Guinea	0.59
Luxembourg	0.51

Year	Control gun ownership	Protect the right of Americans to own guns
1993	0	0
1999	0	0
2003	0	0
2008	58	0
2011	49	50

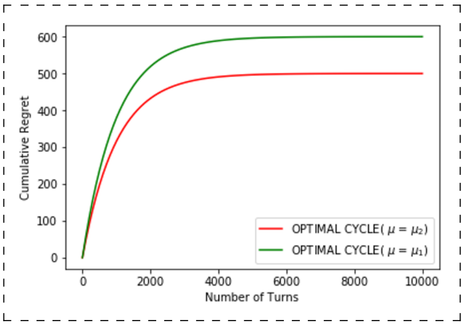
(a) DEPLOT fails to capture the title of the plot.

(b) The query references the color of the bar, which is not captured by the table. However the query is highly contextual.

Figure 5: Select examples from CHARTQA (human) where DEPLOT underperforms with respect to PALI-3.

Query: change of cumulative regret with respect to number of turns where EQUAT-TK, EQUAT-TK, EQUAT-TK.

Chart:



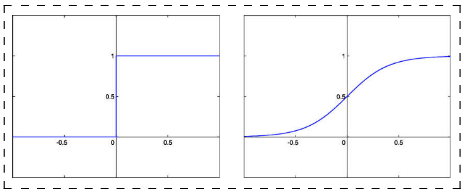
DePlot Table:

TITLE		
Number of Turns	OPTIMAL CYCLE ($\mu = \mu_2$)	OPTIMAL CYCLE ($\mu = \mu_2$)
0	0	0
2000	435	520
4000	490	590
6000	500	600
8000	500	600
10000	500	600

(a) DEPLOT fails to capture the label of the y axis. Here EQUAT-TK is a special token used in SCICAP to replace equations in the caption.

Query: step function and sigmoid function.

Chart:



DePlot Table:

TITLE	
<>	
0	0.5
1	1.0
2	1.0
3	1.0
4	0.5
5	0.9
6	1.0

(b) PALI-3 visually recognises the step and sigmoid functions. DEPLOT fails to handle multiple subplots and outputs a numerical representation that loses the semantic information.

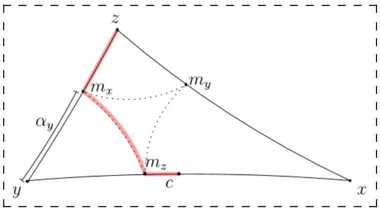
Figure 6: Select examples from SCICAP where DEPLOT underperforms with respect to PALI-3.

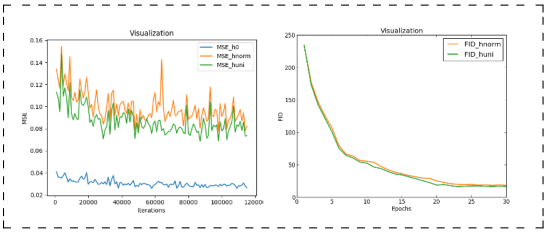
Query

(a) a geodesic triangle Δ (with internal points mx, my, mz and c labelled as in the proof of theorem NUM dashed lines indicate a distance $\leq \delta$ and the red line indicates the upper estimate for d).

(b) illustration of the training process on celeba. left: mean squared errors of the input images and the reconstructions conditioned on different latent codes. right: the fid scores of random generations after each training epoch.

Chart





DePlot Table

(a)

TITLE					
m^3	m_a	m_e	m_e	m_e	m_e
m_a	0	0	0	0	0
m_a	0	0	0	0	0
m_a	0	0	0	0	0
x	0	0	0	0	0
m_a	0	0	0	0	0
last row repeating 19 times...					

(b)

TITLE	Visualization		
Iterations	MSE_h0	MSE_hnorm	MSE_huni
2000	0.04	0.134	0.113
2000	0.036	0.155	0.096
2000	0.032	0.129	0.09
2000	0.031	0.126	0.101
15 more rows with continuing pattern...			

(a) PALI-3 recognises a geodesic triangle. DEPLOT fails to output anything useful as the chart has no underlying table data.

(b) PALI-3 correctly answers a query that refers to visual placement of subplots (left: MSE, right: FID). DEPLOT misses the second subplot completely and spends its output token budget on irrelevant datapoints for the first subplot.

Figure 7: Select examples from SCICAP where DEPLOT underperforms with respect to PALI-3.