Classification-Regression for Chart Comprehension

Matan Levy[1†], Rami Ben-Ari[2], and Dani Lischinski[1]

[1] The Hebrew University of Jerusalem, Israel
[2] OriginAI, Israel

Abstract. Chart question answering (CQA) is a task used for assessing chart comprehension, which is fundamentally different from understanding natural images. CQA requires analyzing the relationships between the textual and the visual components of a chart, in order to answer general questions or infer numerical values. Most existing CQA *datasets* and *models* are based on simplifying assumptions that often enable surpassing human performance. In this work, we address this outcome and propose a new model that jointly learns classification and regression. Our language-vision setup uses co-attention transformers to capture the complex real-world interactions between the question and the textual elements. We validate our design with extensive experiments on the realistic PlotQA dataset, outperforming previous approaches by a large margin, while showing competitive performance on FigureQA. Our model is particularly well suited for realistic questions with out-of-vocabulary answers that require regression.

Keywords: Chart Question Answering, Multimodal Learning

1 Introduction

Figures and charts play a major role in modern communication, help to convey messages by curating data into an easily comprehensible visual form, highlighting the trends and outliers. However, despite tremendous practical importance, chart comprehension has received little attention in the computer vision community. Documents ubiquitously contain a variety of plots. Using computer vision to parse these visualizations can enable extraction of information that cannot be gleaned solely from a document's text. Recently, with the rise of multimodal learning methods, *e.g.*, [4, 6, 18, 21, 23, 25, 26, 30], interest in chart understanding has increased [5, 13–15, 20, 27].

Studies on figure understanding (*e.g.*, [15, 20]), commonly involve answering questions, a task known as Chart Question Answering (CQA). This task is closely related to Visual Question Answering (VQA), which is usually applied on natural

[†]Part of this research was conducted at IBM Research AI, Israel.

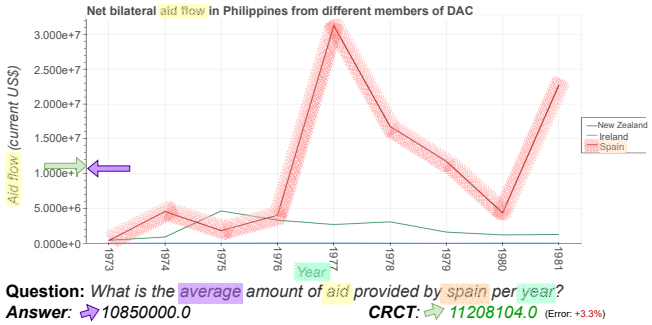


Fig. 1: Interactions marked on a sample from the PlotQA dataset [20], alongside with our CRCT prediction. We highlight the interacting parts/tokens with matching colors. Note the complexity of attention between the different modalities needed to correctly answer the question. The result predicted by CRCT and the ground truth answer are indicated by green and purple arrows.

images [2, 6, 26, 30]. VQA is typically treated as a classification task, where the answer is a category, *e.g.*, [1, 2, 19, 30]. In contrast, answering questions about charts often requires regression. Furthermore, a small local change in a natural image typically has limited effect on the visual recognition outcome, while in a chart, the impact might be extensive. Previous works have demonstrated that standard VQA methods perform poorly on CQA benchmarks [13, 20]. A chart comprehension model must consider the interactions between the question and the various chart elements in order to provide correct answers. The complexity of such interactions is demonstrated in Fig. 1. For example, failing to correctly associate a line with the correct legend text would yield an erroneous answer.

Several previous CQA studies suggest a new dataset along with a new processing model, *e.g.*, [5, 13, 15, 20]. CQA datasets differ in several ways: (1) type and diversity of figures, (2) type and diversity of questions, (3) types of answers (*e.g.*, discrete or continuous). While previous methods have recently reached a saturation level on some datasets, *e.g.*, 94.9% on FigureQA [15], 92.2% on LEAF-QA++ [27], and 97.5% on DVQA [13], Methani *et al.* [20] attribute this to the limitations of these datasets. Hence, they propose a new dataset (PlotQA-D), which is the largest and the most diverse dataset to date, with an order of magnitude more images/figures and $\times 4,000$ different answers. PlotQA-D further contains more challenging and realistic reasoning and data retrieval tasks, with a new model (PlotQA-M) achieving 22.5% accuracy on this dataset, while human performance reached 80.47% [20].

In this paper we further explore the cause behind the saturation of various methods on previous data sets. We argue that similarly to early stages of VQA [8], several common datasets and benchmarks suffer from bias, oversimplicity and classification oriented Q&A, allowing some methods to surpass human performance [14, 27]. Next, we introduce a novel method called Classification - Regression Chart Transformer (CRCT) for CQA. We start with parsing

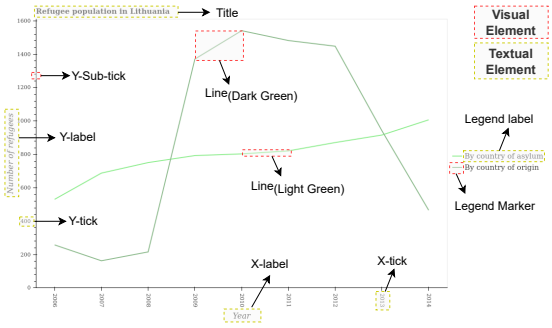


Fig. 2: Examples of object annotations in train images.

the chart with a detector that extracts all of its textual and visual elements, which are then passed, along with the question text, to a dual branch transformer for bimodal learning. Our model features the following novelties: 1) In contrast to previous methods that encode only the question, our language model jointly processes all textual elements in the chart, allowing inter and intra relations between all textual and visual elements. 2) We show high generalization by dropping the common 'string matching' practice (replacing question tokens with certain textual chart elements), and accommodating a co-transformer with pretrained BERT [7]. 3) We introduce a new chart element representation learning, fusing multiple inputs from different domains. 4) Finally, a new hybrid prediction head is suggested, allowing unification of classification and regression into a single model. By jointly optimizing our model end-to-end for all types of questions, we further leverage the multi-task learning regime [31].

We test our model on the challenging and more realistic dataset of PlotQA-D, as well as on FigureQA. Our results show that CRCT outperforms the previous method by a large margin on PlotQA-D (76.94% vs. 53.96% total accuracy), capable of matching previous results with 10% of the training data. We further analyze our model via explainability visualizations, revealing its limitations as well as strong capabilities.

2 Related Work

In this section, we review existing CQA models, while focusing on the datasets in Sec. 3. In particular, we find that previous methods are often over-fitted to the type of datasets and corresponding questions/answers (Q&A).

Some CQA methods take the entire chart image as input to the model [13–15], while others first parse the image to extract visual elements using a detector [5, 20, 27]. An example of chart elements and their corresponding class name, obtained from a detector, are shown in Fig. 2.

The pioneering model of Kahou *et al.* [15] outputs binary (Yes/No) answers using a backbone pretrained on ImageNet fed into a Relation Network (RN) [24],

in parallel to an LSTM [11] used for question encoding. Removing the strong limitation to binary Q&A, Kafle *et al.* [13] proposed a new dataset (DVQA) and a model referred to as SANDY. The dataset introduces new question types with *out-of-vocabulary (OOV)* answers. These answers are chart specific (*e.g.*, *Which item sold the most units in any store?*) and do not necessarily appear in the training set. The SANDY model is a classification network (SAN [30]) with DYnamic encoding. In their approach, each text element in the chart is associated with a unique token in a dynamic encoding dictionary, based on the text location. These elements are then added to the dynamic list of answer classes. Kafle *et al.* [14] later introduced PReFIL, another detector-free model with two branches: a visual branch based on DenseNet [12], and a text branch based on LSTM to encode the question. For bimodal fusion, they apply a series of 1×1 convolutions on concatenated visual and question features.

Singh and Shekar [27] introduced STL-CQA, a new detector-based approach, combining transformers followed by co-transformers [3]. Their method however, relies on replacement of tokens from the question with their string match in the chart, therefore tailored to the dataset question generator and is trained on its dictionary. As also claimed by the authors, STL-CQA is likely to fail in real cases where entities are addressed through their variations, which is the case in a reality as represented also in the PlotQA-D dataset.

All the above methods use only a classification head, without a regression capability, strongly limiting the generalization of these methods to realistic charts. OOV answers are therefore limited only to values appearing in the chart's image or *seen in train set* and added a-priori to the answer classes (see Tab. 1, Sec. 3). They commonly overlook the lingual relations between the chart's text, such as the relations between the content of the title, the legend, and the question. Instead, they only rely on the position of the text in the chart as a hint for its class. Nevertheless, PReFIL showed overall accuracy above 93% on FigureQA and DVQA surpassing human performance. Recent results shown in [20] imply that these datasets are strictly "forgiving" with respect to regression capability and lingual interactions between the questions and chart text (see Sec. 3).

Recently, Methani *et al.* [20] introduced a new method (PlotQA-M) and dataset (PlotQA-D). To the best of our knowledge, this is the first model to address the regression task, suggesting a solution for reasoning on realistic charts. PlotQA-M uses a visual detector and two separate pipelines. In a staging structure, a trained classifier switches between the pipelines, one handling fixed vocabulary classification, and the other for dealing with OOV and regression. In its OOV branch, PlotQA-M first converts the chart to a table and uses a standard table question-answering [22], to generate an answer. This pipeline branching complicates the model requiring each pipeline to be optimized separately and trained on a separate subset of the data, missing the impact of multi-task learning, which we further show as a strong advantage. Furthermore, PlotQA-M inter and intra visual-text interactions from the chart image are only determined through question encoding and a preprocessing stage using prior assumption on proximity between chart elements.

Table 1: CQA datasets comparison. Real world vocabulary refers to axes variables. Some datasets apply question paraphrasing (par.)

Dataset	#Plot types	#Plot images	#Q&A pairs	Avg. question length	Q&A #Templates	#Unique answers	Open vocab.	Real World Vocabulary	Semantic Relations	Bbox Ann.	Regression answers	Publicly Available
FigureQA	4	180k	2.4M	33.39	15 (no variations)	2	✗	✗ (100 colors names)	✗	✓	✗	✓
DVQA	1	300k	3.5M	55.22	26 (w\o par.)	1.5k	✓ (Strings)	✗ (1K nouns)	✗	Partial	✗	✓
LEAF-QA	5	246k	1.9M	-	35 (with par.)	12k	✓ (Strings)	✓	✓	✓	✗	✗
LEAF-QA++	5	246k	2.6M	65.65	75 (with par.)	25k	✓ (Strings)	✓	✓	✓	✗	✗
PlotQA-D1	3	224k	8.2M	78.96	74 (with par.)	1M	✓ (Strings, Floats)	✓	✓	✓	✓ (29.86%)	✓
PlotQA-D2	3	224k	29M	105.18	74 (with par.)	5.7M	✓ (Strings, Floats)	✓	✓	✓	✓ (88.84%)	✓

3 Datasets

In this section we discuss the properties of existing CQA datasets, emphasizing the bias they introduce into the models and the evaluation methodologies that were proposed. Tab. 1 presents various properties of these datasets that may strongly impact the realism and generalization of the results to a real world application. This is an extended version of a table shown by Methani *et al*. [20].

Probably the most popular CQA datasets/benchmarks are FigureQA [15] and DVQA [13], both of which are publicly available. FigureQA consists of line plots, bar charts, pie plots, and dot line plots, with question templates that require binary answers. The plot titles and the axes label strings are constant; the axes range is mostly in $[0, 100]$ with low variation; and the legends are chosen from a small set of *color names* (see example in supplementary material). These properties detract from the realism of this dataset.

DVQA [13] contains a single type of charts (bar charts), but offers more complexity in Q&A. The answers are no longer only binary, and may be out of vocabulary (OOV). Questions are split to three conceptual types: **Structural**, **Data retrieval** and **Reasoning**. Structural questions refer to the chart's structure (*e.g.*, *How many bars are there?*). Data retrieval questions require the retrieval of information from the chart (*e.g.*, *What is the label of the third bar from the bottom?*). Reasoning questions demand a higher level of perceptual understanding from the chart and require a combination of several sub-tasks (*e.g.*, *Which algorithm has the lowest accuracy across all datasets?*). Yet, this dataset suffers from lack of semantic relations between the text elements (*e.g.*, bar and legend labels are randomly selected words), and the range of values on the Y-axis is limited. About 46 out of 1.5K unique answers are numeric, consisting of integers with the same values in the train and test sets, allowing a classification head to handle data retrieval and reasoning.

Two more datasets LEAF-QA [5] and LEAF-QA++ [27], have fewer Q&A pairs than DVQA, but several types of charts, and use a real world vocabulary with semantic relations (see Tab. 1). However, they are both proprietary. All the mentioned datasets share a strong limitation, lack of regression Q&A, indicated by their question templates and their discrete answer set. PlotQA-D [20]

is, however, the largest and most comprehensive publicly released dataset to date. This dataset consists of charts generated from real-world data, thereby exhibiting realistic lingual relations between textual elements. The questions and answers are based on multiple crowd-sourced templates. PlotQA-D consists of three different chart types: line-plots, bar-charts (horizontal and vertical), and dot line plots. The range of the Y-axis values is orders of magnitudes larger (up to $[0, 3.5 \times 10^{15}]$) with non-integer answers generally not seen in training, resulting over 5.7M of different answers. In contrast to previous datasets, PlotQA-D often requires a regressor for correctly answering questions. Nearly 30% and 90% of questions require regression in PlotQA-D1 and PlotQA-D2 respectively (see Tab. 1). To the best of our knowledge, PlotQA-D is currently the most realistic publicly available dataset. PlotQA-D offers two benchmarks, the first version of the dataset PlotQA-D1, and its extended version PlotQA-D2, which contains the former as a subset (28% of the Q&A pairs on the charts). The majority of PlotQA-D2 question types require regression (see the suppl. material). We believe that saturated performance on DVQA (97.5%), probably attributed to a single plot type and having only 1.5K unique in contrast to 5.7M answers in PlotQA-D, makes it inappropriate for regression benchmarking.

4 Method

We present an overview of our CRCT architecture for CQA in Fig. 3. In our approach, the image is first parsed by a trained object detector (see object classes in Fig. 2). The output of the parsing stage are object classes, positions (bounding boxes), and visual features. All of the above are projected into a single representation per visual element, then stacked to form the *visual sequence*. Similarly, each textual element is represented by fusing its text tokens, positional encoding and class. Together with the question text tokens, we obtain the *text sequence*. The two sequences are fed in parallel to a bimodal co-attention-transformer (co-transformer). The output of the co-transformer are pooled visual and textual representations that are then fused by Hadamard product and concatenation, and fed into our unified classification-regression head. In the next sections we describe the train and test configurations in detail.

Visual Encoding: The visual branch encodes all the visual elements in the chart, *e.g.*, line segments or legend markers. For visual encoding we train a Mask-RCNN [9] with a ResNet-50 [10] backbone. Object representations are then extracted from the penultimate layer in the classification branch. In our detection scheme objects are textual elements (*e.g.*, title, xlabel) as well as visual elements (*e.g.*, plot segment) as shown in Fig. 2. We create a single representation per visual element by a learnable block as shown in Fig. 4a. This block takes as input the 4D vector describing the bounding box (normalized top-left and bottom-right coordinates), the class label and the object representation produced by the detector (encapsulating *e.g.*, the line direction), and projects them to an embedding space (1024D).

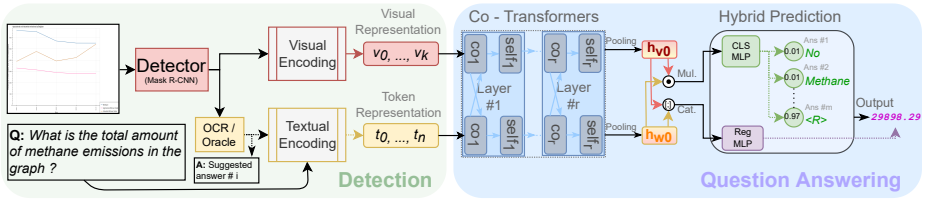
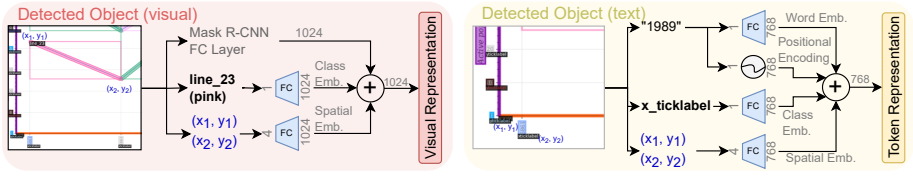


Fig. 3: Our Classification - Regression Chart Transformer (CRCT) network architecture consists of two stages of detection and question answering. The detection stage (left) provides bounding boxes and object representations of the visual and textual elements (see Fig. 2). These features, along with the question text, enable the co-transformers in the second stage (right) to fuse both visual and textual information into a pooled tuple of two single feature vectors $\{\mathbf{h_{v_0}}, \mathbf{h_{w_0}}\}$. Next, our hybrid prediction head containing two different MLPs, outputs a classification score and a regression result. $\text{co}_i/\text{self}_i$: co/self attention.



(a) Visual Representation. **(b)** Textual Representation (per token).

Fig. 4: Chart element representations. The relevant information for representing each type of element is summed into a single vector.

Object colors are generally encoded in the representation output from the detector. However the actual colors are often important for linking the legend marker to the legend label (text), allowing the connection between the question and the target line or bar in the chart. Our observation shows that training the detector with decomposition of graphs to colors, boosts the performance. Finally, our visual element representations form a sequence, is denoted by $v_1, ..., v_k$. We further add the global plot representation (v_0) as [CLS] token.

Text Encoding: Raw text is handled with a pretrained BERT [7]. The textual features are derived from the question and the text contained within the chart, such as the axes labels, legends and title. In contrast to VQA where the lingual part includes only the question, in CQA there are additional text elements that are essential for chart comprehension. Text position in the chart carries important information. In this study, we encode the textual elements in a concatenated version, separated with the special [SEP] token, followed by the question and an answer with the special token [CLS] on top (t_0). In contrast to previous work [13, 15, 18, 27, 30], where only the question (or question + answer) was encoded, here the text encoder is generalized to include all textual elements enriched with their spatial location and class. This approach allows free data-

driven interaction between different visual and textual elements, *e.g.*, the legend marker and its corresponding text, as well as interactions between text sub-elements, *e.g.*, the answer and *part* of the Y-axis label or title. To this end, we create a new representation from all the textual elements in the chart by fusing the word embedding, the positional encoding, the text location in the chart and the text class embedding. This fusion is carried out through a MLP layer, including projection and summation as shown in Fig. 4b.

4.1 Associating Visual and Textual Elements

For multi-modal interaction we rely on the co-attention architecture that was first suggested for machine translation in [3]. This model contains two different sequence to sequence branches: visual and textual, as shown in Fig. 3. The information in the two streams is fused through a set of attention block exchanges, called co-attention. We use a transformer with 6 blocks of two encoders with *co-* and *self-* attention. Each encoder computes a query Q, key K, and value V matrices, followed by feed-forward layer, skip connections and normalization [28]. In order to exchange the information between the modalities, the *co*-transformer's keys and values at each stream are mutually exchanged resulting a cross-modality attention. Finally, the resulting $\{\mathbf{h_{v_0}}, \mathbf{h_{w_0}}\}$ pooling tokens (indicated by [CLS] special token) are forwarded to the classification and regression heads (see Fig. 3). For more details, see suppl. material.

4.2 Question Answering Stage

Similar to previous work [5, 13, 14, 20, 27] and in order to allow fair comparison, we use an oracle to recognize the extracted text elements. The oracle is a perfect text recognition machine, and is used to disentangle the impact of OCR accuracy. Previous work frequently assume a perfect text detector, *e.g.*, [13,15,20,27]. In this work however, we explicitly account for inaccuracies in the detector by considering only text elements from the oracle with $IoU > 0.5$. We then create the set of possible answers for classification, composed of *in-vocabulary* (*e.g.*, Yes / No) and *out-of-vocabulary* (OOV) answers (*e.g.*, the title or specific legend label). OOV additional classes (dynamically added) allow dealing with chart specific answers that has not been seen during training. To predict the correct answer, we train the model with binary cross-entropy loss. To this end, we concatenate the answer to the question in the textual branch, pass it through the model and evaluate a score in $[0, 1]$ range (see Fig. 3). This score indicates the model's certainty whether the answer is aligned with the question (correct) or not (wrong).

4.3 Unified Prediction

Previous works frequently use only a classification head, overlooking regression [5, 13, 15, 27], or use a totally separate pipeline for the regression task [20]. In

classification based methods, the answers are restricted to discrete values, that are part of the numeric values appearing on the chart. This approach strongly limits the generalization, lacking the capability to predict unseen numeric values or charts with unseen ranges. In this work, we propose a novel hybrid prediction head allowing unified classification-regression. To this end, we add a regression soft decision flag $\langle R \rangle$ as an answer class, followed by a regressor. During training the model learns which type of questions require regression by choosing the $\langle R \rangle$ class as the correct answer. A separate and consequent regression is then applied to generate the answer (see Fig. 3). Note that during training, the loss changes dynamically from BCE loss for classification and L1 loss for regression, so the network is jointly optimized for classification and regression. During train, we vanish the regression loss when the correct class is not $\langle R \rangle$. The hybrid prediction allows joint training on all types of Q&As, leveraging multi-task learning.

4.4 Implementation Details

For training the CRCT we use two stages. We first train a Mask-RCNN [9] from which the visual features are derived, using Detectron2 [29] library. We then train the co-transformer model for 20 epochs with linear learning rate scheduler. We use binary cross entropy loss for the classification component and L1 loss for regression. For answer alignment prediction (as described in Sec. 4.2), we generate negative examples by randomly assigning wrong answers to questions. Training our model on PlotQA-D1 took 3.5 days on two Nvidia RTX-6000 GPUs. The inference computational cost is proportional to the size of candidate answers. In our experiments the inference time took 0.23 seconds per question. Our code and models are publicly available at https://github.com/levymsn/CQA-CRCT.

5 Evaluation

As evaluation benchmark we opted for PlotQA-D and FigureQA datasets, being fully annotated to train a detector (DVQA lacks the important annotation of legend markers). Yet, we focus our analysis on PlotQA-D for several reasons: (1) Publicly available to allow benchmarking. (2) The scale: Over $\times 10$ larger Q&A pairs and over $\times 1000$ more unique answers, than the predecessors (see Tab. 1); (3) Highly variable axis scale; (4) Having diverse and realistic questions/answers with rich vocabulary titles, legend labels, X and Y labels including initials gathered from real figures; (5) Most importantly, question types that require regression and therefore reflect a realistic case for CQA.

In terms of methods to compare with, we searched for publicly available code or assessments on the chosen datasets. To allow a fair comparison to previous methods, in addition to PlotQA-M, we further test PReFIL [14] on PlotQA-D. To this end, we trained PReFIL on PlotQA-D1. We chose PreFIL due to it's high performance on DVQA and FigureQA and as a representative candidate for previous methods that rely on classification and lack a regression capability. Since PReFIL has only a classification head we quantized the numeric values

into Y-ticks and added them to the dynamic classification head in training and also at test (a common practice, also performed in PReFIL [14]). For sake of analysis and to allow a fair comparison we show the PReFIL results for numeric evaluation with various error tolerances (see Fig. 5b).

To handle the wide range of Y-axis values in PlotQA-D, we normalize values to $[-1, 1]$ (by detecting X-Y axes and their values). This improves convergence and enables scale invariant prediction. We output answers in the same range.

5.1 Results

We train our model on PlotQA-D1 dataset, that consists of one third of PlotQA-D2 in questions, while testing on both PlotQA-D1 and PlotQA-D2 test sets. We show significant improvements on both test sets. Results are shown as average accuracy over the test set and accuracy breakdown per-question category.

Comparison to previous methods: Tab. 2a summarizes the results on PlotQA-D1 test set. In general, we outperform PlotQA-M in all categories by a large margin. For instance, the gaps for Data Retrieval and Reasoning are 48.8% (94.52% vs. 45.68%) and 23.7% (54.87% vs. 31.20%) *absolute* points, respectively. Finally, on average we achieve 76.94% accuracy, compared to 53.96% of PlotQA-M. While outperforming PlotQA-M when trained on the same train set, in the next experiment we show the extent of train data reduction that can be allowed to match the previous results of PlotQA-M. This experiment shows that as little as 10% of training data (randomly selected) are already sufficient to reach this goal. (see CRCT-10% in 2a).

With respect to PReFIL, while we show comparable results on the Structural question category, containing classification type questions, CRCT is superior to PReFIL in all other categories. As expected, PReFIL performs poorly on Data Retrieval and particularly Reasoning Q&As (only 31.66% vs 54.87% for our CRCT) due to lack of regression capability. In total average accuracy we surpass both PlotQA-M and PReFIL by 23% and 19% *absolute* points, respectively. Interestingly, with our quantization scheme training of PReFIL, it outperforms PlotQA-M, in all categories.

Due to extreme computational demand for train on PlotQA-D2, in the next experiment we train PReFIL and CRCT on PlotQA-D1 train set and report the results on PlotQA-D2 test set in Tab. 2b. Note that for PlotQA-M we report the result from [20] with the model trained on whole PlotQA-D2. These results show that even when we train on PlotQA-D1 dataset we are able to outperform PlotQA-M trained on $\times 3$ larger size data, in all categories, often with significant margin. Our CRCT is superior here also to PReFIL with average accuracy of 34.44% vs. 10.37%. Note the poor performance of PReFIL on Reasoning category, from which many questions require regression, reaching 3.9% comparing 25.81% in CRCT. These results show the significance of our hybrid classification-regression capability.

Regression Performance: The accuracy of regression errors are often measured by L_2 or L_1 differences or by ER-error rate. In PlotQA-D [20], a regression answer is considered correct if it falls within $\pm 5\%$ tolerance from the ground

truth value. This measure, however, is proportional to the true value, vanishing (no tolerance) for true values near zero. We therefore suggest the *tick-based* error measure as more appropriate for extraction of numerical values. To this end we suggest a constant gap per-chart, defined as a fraction of units between two consecutive sub-ticks (see Fig. 2) *e.g.*, 1/4 sub-tick.

In PlotQA-D1, 29% of the questions require regression. Following PlotQA [20], we show in Fig. 5a CRCT accuracy distribution considering the error rate (ER) measure. We observe that 44.37% of the answers are within ±5% of the true value. The prevalence of errors decreases in higher tolerance ranges except the *outlier* in the tail, indicating that 11.3% of the answers were over 100% off the true value. As expected, we observe that CRCT error distribution indeed accumulates near zero true values (see suppl. material), justifying the advantage of value invariant error measure. Fig. 5b shows the variation of regression accuracy with increased tolerance (as sub-tick fraction) for CRCT and PReFIL. CRCT achieves over 85% total accuracy and 78% regression accuracy for 1 sub-tick tolerance. Note the large gap w.r.t PReFIL through all the range as well as the drop in CRCT-10% that obtained similar accuracy to PlotQA-M (Fig. 5a). For visual examples of our CRCT model on regression assignment see Figures 1, 6b and the suppl. material.

Results on FigureQA: Although our model's strength is in general Q&A with regression, we also test our model on the binary answer data set of FigureQA [15]. FigureQA's training set was generated using different 100 colors. This dataset contains two families of validation and test sets. The first family is the Val-1/Test-1 sets, that was generated using the original color schemes as in the train set. On the contrary, Val-2/Test-2 sets consist of alternate color scheme that *was not seen* in the train set at all. Tab. 3 presents a comparison on FigureQA dataset. CRCT shows comparable performance to SoTA on the original color scheme. While we outperform previous methods on the alternate color scheme sets, we reach an inferior performance w.r.t PReFIL. This test indicates a color sensitivity for our detector-based approach as we discuss in Sec. 8.

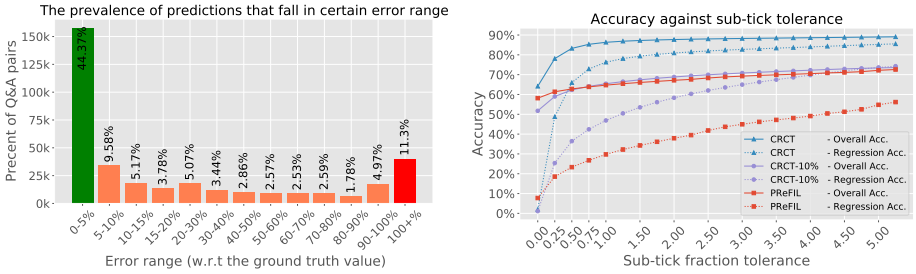
Table 2: Accuracies [%] on PlotQA test sets. Values in each column indicate average accuracy per-question category. CRCT and PReFIL are trained on the PlotQA-D1 subset. PReFIL results are reproduced. 'CRCT-10%' indicates our results with training on 10% of the PlotQA-D1 train set. S, D and R stand for Structural, Data Retrieval and Reasoning question categories, respectively

(a) Evaluation on PlotQA-D1 test set

Method	S	D	R	Overall
PlotQA-M [20]	86.31	45.68	31.2	53.96
CRCT-10%	87.15	74.71	29.19	57.75
PReFIL [14]	**96.66**	58.69	31.66	57.91
CRCT (ours)	96.13	**94.52**	**54.87**	**76.94**

(b) Evaluation on PlotQA-D2 test set

Method	S	D	R	Overall
PReFIL [14]	**96.66**	21.9	3.9	10.37
PlotQA-M [20]	75.99	58.94	15.77	22.52
CRCT (ours)	96.23	**66.65**	**25.81**	**34.44**



(a) The prevalence of CRCT's answers that fall in certain error range.

(b) Accuracy for different sub-tick error range (tolerance).

Fig. 5: Model regressor performances on PlotQA-D1. In 5a, the green column shows the "correct" answers *i.e.* fall in 5% tolerance. 11.3% of the answers (red) miss the target by more than 100%. In 5b, $x = 0$ indicates exact match between prediction and ground truth (zero tolerance).

Table 3: Accuracy on FigureQA dataset [15]. Second place is coloured in brown

(a) Original color scheme

Model / Acc.	Val.	Test
RN [15]	-	76.52
LEAF-Net [5]	-	-
Zou et al. [32]	85.48	85.37
CRCT (ours)	**94.61**	**94.23**
PReFIL [14]	**94.84**	**94.88**

(b) Alternate color scheme

Model / Acc.	Val.	Test
RN [15]	72.54	72.40
LEAF-Net [5]	81.15	-
Zou et al. [32]	82.95	83.05
CRCT (ours)	**85.04**	**84.77**
PReFIL [14]	**93.26**	**93.16**

6 Ablation Study

Tab. 4 shows an ablation study of our method using different configurations. First we examine the impact of the *legend marker* (see Fig. 2) as key element. Removing it from the input in the visual branch prevents the model to associate the question to the specific plots/bar in multi-graph chart. The results show drop in performance in all categories with total accuracy dropping from 57.75% to 50.45%. In the next two tests we show the impact of representation architecture on the end results. To this end we remove the class label embeddings from the visual and textual representation (*e.g.*, 'line_23' or 'x_ticklabel' in Fig. 4). Although noisy, these inputs derived from the detector, positively impact the results. Removing them, causes regression accuracy to drop from 20.74% to 17.35%, for visual and 15.51% for text. We observe the best classification performance is achieved without the visual class embedding. However, this embedding is just one component of the visual representation (see Fig. 4a - Class-Emb). In some cases Class-Emb is redundant to the visual representation, and removing it can slightly improve certain classification Q&As, resulting in this outcome (*e.g.*, where only textual elements are addressed). However, as Tab. 4 shows, the slight improvement in classification task ($\sim$ 1%) is traded

Table 4: Ablation study with different configurations (see also Fig. 4). All models are trained on 10% of PlotQA-D1 train set, and evaluated on the entire PlotQA-D1 test set. S, D and R stand for Structural, Data Retrieval and Reasoning, respectively

Method	Regression	Classification	S	D	R	Overall
w/o Legend Marker	14.76	65.02	81.13	56.01	27.05	50.45
w/o *Textual* Class Emb.	15.51	66.86	81.75	61.73	26.96	51.98
w/o *Visual* Class Emb.	17.35	73.68	85.06	73.09	30.57	57.36
Only Bbox for *Visual* Feats.	18.66	68.68	84.97	72.94	23.75	54.19
Two Pipelines	14.80	70.19	84.49	68.65	25.16	53.64
CRCT	20.74	72.86	87.15	74.71	29.19	**57.75**

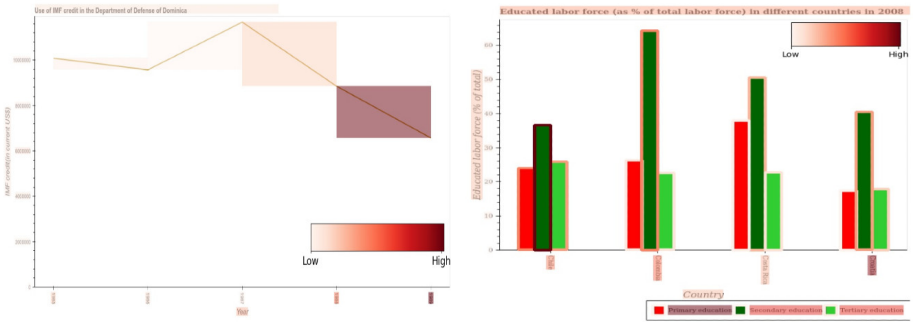
with large degradation in regression accuracy ($\sim 3\%$), resulting a lower total accuracy. When removing all features except the bounding box coordinates, from the visual representation, the total accuracy drops by 3.6%. This shows the importance of all elements in our chart element representation model (see Fig. 4). Finally, we examine the importance of the multi-tasking regime inherent in our unified classification-regression network. To this end we train our classification and regression network separately (similar to [20]). Assuming an oracle for routing classification and regression type questions to the proper network, we report the outcome accuracies. We observe performance drop on all categories emphasizing the importance of combining both regression and classification in CRCT's learning process. Our detector achieved AP50=0.90. Testing our model with ground truth detections had a negligible effect on the accuracy.

7 Explainability

We provide visualizations for CRCT attention using the *Captum* package [16,17]. Often, relatively few units in a NN are highly influential towards a particular class [17]. Considering the true answer, we integrate over the input gradients to find the most influential features. We then color code the image to indicate the regions in the chart, visual or textual, that the network found influential in answering the posed question. Fig. 6 shows such visualization maps over charts, on examples from the test set. In Fig. 6a CRCT correctly "looks" at the x-tick at the global minimum in the plot and on the corresponding x-label, when asked about the *minimum* argument. Fig. 6b shows an example of a bar chart. Note that CRCT's attention is driven toward the dark-green bars due to the question asking about the *average* for a certain category (*secondary education*). As observed, CRCT attends intuitive features and spatial locations according to the questions asked. For more examples see the suppl. material.

8 Summary and Discussion

In this paper we argue that the simplicity of Chart Question Answering (CQA) associated with lack of realistic chart content and question types, has lead pre-



(a) Q: *In which year was the use of IMF credit in DoD minimum?*
GT: 1989, CRCT: 1989.

(b) Q: *What is the average percentage of labor force who received secondary education per country?*
GT: 48.05, CRCT: 47.91 (Error: -0.29%).

Fig. 6: Test set visualizations. Warmer box color means higher influence.

vious methods to omit the regression task. The recent PlotQA work [20] addresses these shortcomings, suggesting a remedy via a new large scale and diverse dataset, as well as a new model. We hereby suggest a bimodal framework for CQA that leverages the natural lingual inter-relations between different chart elements and introduce a novel unified classification-regression head. Our explainability visualizations shed light on question-chart understanding of our model.

We evaluate our method on the PlotQA and FigureQA datasets, significantly outperforming the PlotQA model. We further compare our method to a previous classification based method of PReFIL, that reached SoTA results on FigureQA (also high performing on DVQA) observing a strong drop in performance when tested on more challenging datasets such as PlotQA-D. We argue that the edge of our method is not in classification but rather on the combined classification regression tasks with natural lingual relations that exist in real CQA case.

However, some limitations still remains, such as sensitivity to color combinations and non-linear axis scales. Although we reach a comparable result to PReFIL on FigureQA, we noticed deterioration in results when the test and train colors are different. We relate this limitation to the detector representation learning, including the color attributes from the charts and relying on them to distinguish between the plots in a chart. In practice, this limitation can be overcome by extending the (synthetic) dataset to contain more colors.

In future work we intend to relax the need for full chart annotations, and tackle the efficiency of the training. With PlotQA opening the door again toward chasing human performance in chart comprehension, we hope this paper will encourage researchers to take this challenge.

Acknowledgments: We thank Or Kedar and Nir Zabari for their assistance in parts of this research. We thank PlotQA [20] authors for sharing additional breakdowns. This work was supported in part by the Israel Science Foundation (grant 2492/20).

References

1. Acharya, M., Kafle, K., Kanan, C.: TallyQA: Answering Complex Counting Questions. In: AAAI (2019) 2
2. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: VQA: Visual Question Answering. In: ICCV (2015) 2
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. In: ICLR (2015) 4, 8
4. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In: CVPR (2021) 1
5. Chaudhry, R., Shekhar, S., Gupta, U., Maneriker, P., Bansal, P., Joshi, A.: LEAF-QA: Locate, Encode & Attend for Figure Question Answering. In: WACV (2020) 1, 2, 3, 5, 8, 12
6. Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J.M., Parikh, D., Batra, D.: Visual Dialog. In: CVPR (2017) 1, 2
7. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL-HLT (2019) 3, 7
8. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In: CVPR 2017 (2017) 2
9. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017) 6, 9
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR (2016) 6
11. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation (1997) 4
12. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely Connected Convolutional Networks. In: CVPR (2017) 4
13. Kafle, K., Cohen, S., Price, B., Kanan, C.: DVQA: Understanding Data Visualizations via Question Answering. In: CVPR (2018) 1, 2, 3, 4, 5, 7, 8
14. Kafle, K., Shrestha, R., Cohen, S., Price, B., Kanan, C.: Answering questions about data visualizations using efficient bimodal fusion. In: WACV (2020) 1, 2, 3, 4, 8, 9, 10, 11, 12
15. Kahou, S.E., Michalski, V., Atkinson, A., Kádár, Á., Trischler, A., Bengio, Y.: FigureQA: An Annotated Figure Dataset for Visual Reasoning. In: ICLRW (2018) 1, 2, 3, 5, 7, 8, 11, 12
16. Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Reynolds, J., Melnikov, A., Lunova, N., Reblitz-Richardson, O.: PyTorch Captum. https://github.com/pytorch/captum (2019) 13
17. Leino, K., Sen, S., Datta, A., Fredrikson, M., Li, L.: Influence-Directed Explanations for Deep Convolutional Networks. In: ITC (2018) 13
18. Lu, J., Batra, D., Parikh, D., Lee, S.: ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In: NeurIPS. pp. 13–23 (2019) 1, 7
19. Malinowski, M., Fritz, M.: A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input. In: NeurIPS (2014) 2
20. Methani, N., Ganguly, P., Khapra, M.M., Kumar, P.: PlotQA: Reasoning over Scientific Plots. In: WACV (March 2020) 1, 2, 3, 4, 5, 8, 10, 11, 13, 14

21. Miech, A., Zhukov, D., Alayrac, J., Tapaswi, M., Laptev, I., Sivic, J.: HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In: ICCV (2019) 1
22. Pasupat, P., Liang, P.: Compositional Semantic Parsing on Semi-Structured Tables. In: ACL (Jul 2015) 4
23. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning Transferable Visual Models From Natural Language Supervision. In: Meila, M., Zhang, T. (eds.) ICML (2021) 1
24. Santoro, A., Raposo, D., Barrett, D.G.T., Malinowski, M., Pascanu, R., Battaglia, P.W., Lillicrap, T.: A simple neural network module for relational reasoning. In: NIPS (2017) 3
25. Schwartz, I., Yu, S., Hazan, T., Schwing, A.G.: Factor Graph Attention. In: CVPR (2019) 1
26. Singh, A., Natarjan, V., Shah, M., Jiang, Y., Chen, X., Parikh, D., Rohrbach, M.: Towards VQA Models That Can Read. In: CVPR (2019) 1, 2
27. Singh, H., Shekhar, S.: STL-CQA: Structure-based Transformers with Localization and Encoding for Chart Question Answering. In: EMNLP (Nov 2020) 1, 2, 3, 4, 5, 7, 8
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need. In: NeurIPS (2017) 8
29. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019) 9
30. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.J.: Stacked Attention Networks for Image Question Answering. In: CVPR (2016) 1, 2, 4, 7
31. Zhang, Y., Yang, Q.: A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering (2021) 3
32. Zou, J., Wu, G., Xue, T., Wu, Q.: An Affinity-Driven Relation Network for Figure Question Answering. 2020 ICME (2020) 12