Answering Questions about Data Visualizations using Efficient Bimodal Fusion

Kushal Kafle[1] Robik Shrestha[1] Brian Price[2] Scott Cohen[2] Christopher Kanan[1,3,4]
[1]Rochester Institute of Technology [2]Adobe Research [3]Paige [4]Cornell Tech
[1]{kk6055, rss9369, kanan}@rit.edu [2]{bprice, scohen}@adobe.com

Abstract

Chart question answering (CQA) is a newly proposed visual question answering (VQA) task where an algorithm must answer questions about data visualizations, e.g. bar charts, pie charts, and line graphs. CQA requires capabilities that natural-image VQA algorithms lack: fine-grained measurements, optical character recognition, and handling out-of-vocabulary words in both questions and answers. Without modifications, state-of-the-art VQA algorithms perform poorly on this task. Here, we propose a novel CQA algorithm called parallel recurrent fusion of image and language (PReFIL). PReFIL first learns bimodal embeddings by fusing question and image features and then intelligently aggregates these learned embeddings to answer the given question. Despite its simplicity, PReFIL greatly surpasses state-of-the art systems and human baselines on both the FigureQA and DVQA datasets. Additionally, we demonstrate that PReFIL can be used to reconstruct tables by asking a series of questions about a chart.

1. Introduction

Data visualizations such as bar charts, pie charts, and line graphs are common ways to present complex data in a manner that is easily interpretable to people. They are ubiquitous in both scientific and business documents. Data visualizations are designed to be effective at conveying trends and comparisons in a glance, while also preserving salient details. Using computer vision to parse these visualizations can enable extraction of information that cannot be gleaned by solely studying a document's text. Despite the high potential payoff and tremendous practical value, this problem has received little attention until recently. In 2018, two datasets for answering questions about data visualizations were introduced along with new algorithms [16, 21]; however, there is considerable room for improvement. Here, we propose a novel algorithm that exceeds the state-of-the-art on both of these datasets by a large margin.

Visual question answering (VQA) requires a system to answer questions about images [7, 30, 18, 20]. Several datasets for VQA has been proposed in recent years, include natural image understanding [30, 7], counting [2], reasoning about synthetic scenes [14], medical image analysis [28], scene text understanding [38], and video comprehension [13]. Chart QA (CQA) is a VQA task involving answering questions about data visualizations. Formally, given an data visualization image I and a question Q about I, a CQA model must predict the answer A. CQA requires understanding of the relationships among different 'symbols' (elements in the chart) in an image. In contrast to natural images, even tiny modifications to the image can cause drastic changes in the correct answer, making CQA an excellent platform for studying reasoning mechanisms [21, 16]. CQA often requires optical character recognition (OCR) and handling words unique to a given visualization.

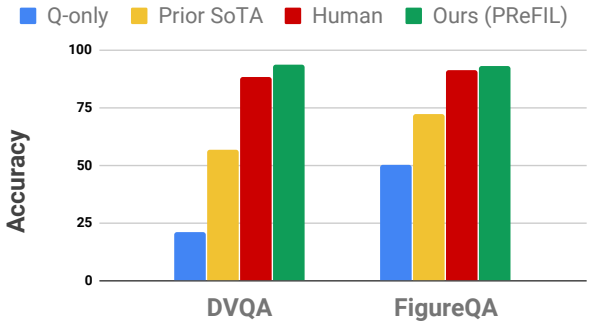


Figure 1. We propose the PReFIL algorithm for chart question answering (CQA). PReFIL surpasses the prior state-of-the-art (SoTA) and human baselines on DVQA and FigureQA datasets.

In this paper, we describe a novel algorithm called parallel recurrent fusion of image and language (PReFIL). PReFIL jointly learns bimodal embeddings by using both low- and high-level image features, which enable it to answer complex questions requiring multi-step reasoning and comparison without employing specialized relational or attention modules. Extensive experiments show that our algorithm outperforms current state-of-the-art methods, by a large margin in two challenging CQA datasets.

Our key contributions are:

- We critically review existing CQA datasets outlining their strengths and weaknesses (Sec. 2.1).
- We collect human performance values for the DVQA dataset using crowd-sourcing (Sec. 4).
- We propose a novel algorithm called parallel recurrent early fusion of image and language (PReFIL) (Sec. 3). PReFIL greatly surpasses existing methods on CQA datasets and also outperforms humans on both DVQA and FigureQA (Sec. 4). PReFIL's code and pre-trained models will be publicly released.
- We pioneer the use of iterative question answering to reconstruct tables from charts (Sec. 4.4).
- In light of our results, we outline a road map toward creating more challenging datasets and algorithms for understanding data visualizations (Sec. 5).

2. Related Work

CQA is a form of VQA. Multiple natural image VQA datasets have been publicly released [30, 7, 34, 27, 17]. VQA has been explored in open-ended [7, 18], counting [2], multiple choice [7, 27], and pointing type setups [43, 1]. Most algorithms treat VQA as a classification problem in which the answer is a category [18]. Several studies have shown that early natural image VQA datasets suffer from a high amount of bias, potentially making it easier for an algorithm to guess the answer without actually understanding of visual content [17, 3, 4, 19]. As a remedy, some subsequent datasets have focused on synthetic scenes and diagrams where reasoning capacities can be better studied [6, 14, 24, 25].

CQA requires capabilities not tested by other VQA tasks due to the innate differences in how information is presented in data visualizations [16, 21]. For instance, the information in charts is conveyed by only a small number of visual elements. Changes to even small image region (e.g., changing color of a legend entry) can drastically alter the information content of the whole chart whereas small changes in a natural image usually affects only a local region. This is one reason why algorithms designed for natural VQA have considerable difficulty when answering questions about data visualizations [16, 21].

Another line of related work involves parsing of visual information in data visualization and other non-natural diagrams. There is a sizable body of prior work in this domain, ranging from extraction of visual elements in a chart [32, 39] to the extraction of underlying data [36, 22, 9]. However, very limited work has been done in a question answering framework where multiple underlying abilities can be represented as a single task.

2.1. Datasets for CQA

Two CQA datasets: DVQA [16] and FigureQA [21], are publicly available at the time of writing this paper. See Table 1 for their statistics. Example images are shown in Fig. 2. We briefly describe and compare both datasets.

DVQA has over 3 million question answer pairs for 300,000 images for bar charts. The question answer pairs in DVQA are divided into three categories: 1) structure understanding (e.g. "How many bars are there?"), 2) data query (e.g., "How many units of item X were sold?"), and 3) reasoning (e.g. "Is the accuracy of algorithm X greater than algorithm Y?"). Since many questions refer to texts specific to the corresponding charts, systems must integrate OCR and dynamically expand their vocabulary to correctly answer questions. DVQA has two test splits: Test-Familiar and Test-Novel, with Test-Novel containing charts with texts that were not seen during training.

FigureQA has over 2 million question answer pairs for 180,000 images. It has five kinds of visualizations: 1) vertical bar charts, 2) horizontal bar charts, 3) pie charts, 4) line graphs and 5) dot-line graphs. Chart element colors are uniformly distributed in the training and validation sets. FigureQA has harder versions of the validation and test sets with color combinations that are unseen in the training set. Validation 1 and Test 1 have the same colors as the training set and Validation 2 and Test 2 have a color scheme that differs from training. Test set annotations are not publicly available. All questions are binary (yes/no) and demand multiple abilities, including finding the largest/smallest element (e.g. "Is X the largest/smallest?"), comparing values of two elements (e.g. "Is X greater/smaller than Y?"), and other scientific measurements (e.g. "Does X have maximum area under the curve?").

2.1.1 DVQA versus FigureQA

DVQA and FigureQA each have their own strengths and shortcomings. We compare and contrast them below.

Shared strengths: Both datasets are large and provide enough training samples to train large scale models, e.g. in DVQA, each unique visual element is repeated at least 1,000 times. Both datasets provide detailed annotations for all figure elements in addition to the question answer pairs, making it possible to create auxiliary tasks or use them as additional training signals. The creators of both datasets tried to eliminate some sources of bias. DVQA has randomized visual elements and it also has a balanced question answer distribution to make guessing difficult. Similarly, FigureQA has a randomized distribution of colors and a balanced distribution of "yes" and "no" answers for each unique question template. Lastly, both datasets provide both easy and hard test splits, where the hard test split measures generalization beyond what is seen during train-

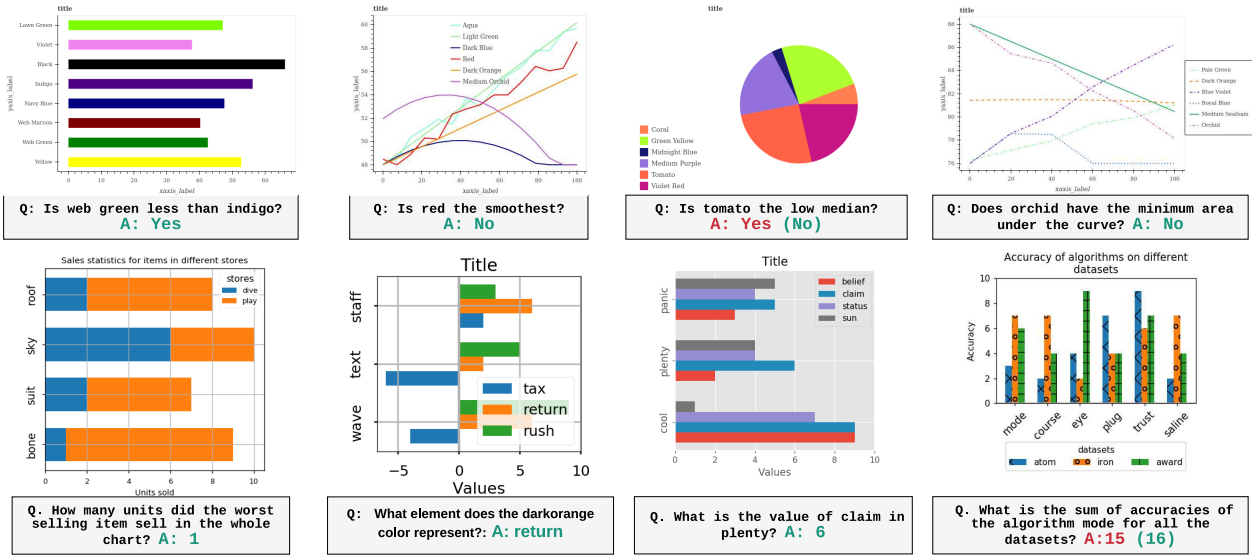


Figure 2. Example images and PReFIL outputs for FigureQA (top) and DVQA (bottom). Red denotes incorrect predictions. For incorrect predictions, correct answer is shown in parentheses. More examples are included in the supplementary materials.

Table 1. FigureQA vs. DVQA

6	Num. Images	Num. QA Pairs	Question Format	Chart Types	Number of Templates	OCR	OOV
DVQA	300,000	3,487,194	Open-ended	1	26 (Plus variations)	Required	Required
FigureQA	180,000	2,38,8698	Yes/No	5	15 (No variations)	Not Required	Not Required

ing. DVQA's "Test Novel" split measures generalization to unseen words and FigureQA provides an "alternated colors" split where visual elements in the chart have different colors than the ones seen during training.

DVQA's advantages: In DVQA, questions about bars are asked by referring to their text labels, e.g. "What is the value of algorithm X?" where X is an actual label in the chart and it will be different for each chart even if they have the same appearance, e.g. identical red bars may have label X in one image and Y in another. This requires integrating OCR into the system. In contrast, FigureQA refers to chart elements by their color, e.g. "red bars" will always be referred to as "red" making it easier for systems to identify a chart's elements. Since DVQA uses chart labels, algorithms must take into account that some of the words may be out-of-vocabulary (OOV) and unseen during training for both questions and answer. To handle this, systems need to have a vocabulary that can be dynamically adjusted during testing. FigureQA has no OOV answers. DVQA also tests for more tasks than FigureQA. For bar charts, DVQA contain most of the tasks in FigureQA (e.g. identifying colors, comparing values, etc.) and several that are not required for FigureQA (e.g. data measurement and inferring structure of the chart). Finally, while DVQA contains only bar charts, its bar charts have increased visual complexity compared to those in FigureQA. FigureQA is limited to single-variable vertical and horizontal bar charts, whereas DVQA also has grouped bar charts and stacked bar charts with legends. DVQA's bars can be hatched, monochrome, and have negative values, all of which are absent in FigureQA.

FigureQA's advantages: While DVQA has only bar charts, FigureQA has three kinds of data visualizations: bar charts, pie charts, and line graphs. This allows FigureQA to have unique question-types that are not encountered for bar chart alone. E.g., for line graphs, FigureQA requires determining the area under the curve, and whether one line intersects another. These are not tested in DVQA. FigureQA also tests compositional reasoning by asking questions about unknown color combinations in chart elements, whereas colors are randomly distributed in DVQA.

Shared limitations: As synthetically generated datasets, both DVQA and FigureQA omit much of the variability found in real-world data visualizations. All of DVQA's charts were made with Matplotlib and all of FigureQA's were made with Bokeh. The variation introduced is limited to the capabilities of these packages. FigureQA uses only generic titles and other chart elements. DVQA has some variety but ultimately is limited to a few templates. Likewise, both datasets have formulaic, templated questions. While questions can be complex, they lack the diversity of human generated queries. In the discussion we elaborate further on how future datasets could overcome these limitations.

2.2. Existing CQA Algorithms

For DVQA, in [16] SANDY (SAN with DYnamic encoding) model was proposed. SANDY used a modified version of the stacked attention network (SAN) [40, 23], which has been widely used for VQA [23, 5]. SAN uses the question to apply attention to the convolutional feature maps. It cannot handle DVQA's OOV words in its test set or the chart specific words found in its questions and answers. To address this, SANDY uses an off-the-shelf OCR method to recognize such words and introduced dynamic encoding to represent OOV and chart-specific words. SANDY's dynamic encoding scheme for OCR can be incorporated into any classification-based VQA algorithm.

FigureQA's creators used a relation network (RN) [35] on their dataset. RN encodes pairwise interactions between every pair of "objects" in an image, enabling it to answer questions involving relationships. Each "object" is a cell of a convolutional feature map. RN has been shown to be especially effective at compositional reasoning in CLEVR [35], and it exceeded baselines on FigureQA.

FigureNet [33] is a multi-step algorithm for FigureQA composed of different modules. The first module is called the spectral segregator, which identifies the elements and colors of the chart. It is followed by the extraction module, which quantifies the values represented by each element. This is then used with a feed-forward network to predict the answer. FigureNet uses the detailed annotations of FigureQA's chart elements to pre-train each of the modules. Because FigureNet relies on having access to the measurements of each chart element, they could only apply it to FigureNet's bar and pie charts.

To assess bias in their datasets, the creators of FigureQA and DVQA both studied question-blind and image-blind models. They found that these models performed abysmally indicating that vision and language must be jointly used to correctly answer the questions. The creators of both datasets also tested simple question+image fusion schemes. These worked better than the blind baselines, but this did not suffice for handling the complexity found in CQA. This is in contrast to VQA with natural images, where these algorithms fare comparatively well.

Compared to existing work, our model does not employ complex attention or relational modules, and unlike FigureNet, it does not require additional supervised annotations for training on FigureQA.

3. The PReFIL Model

We propose the PReFIL algorithm for CQA. As shown in Fig. 3, PReFIL has two parallel Q+I fusion branches. Each branch takes in question features (from an LSTM) and image features from two locations of a 40-layer DenseNet, *i.e.* low-level features (from layer 14) and high-level features (from layer 40). Each Q+I fusion block concatenates the question features to each element of the convolutional feature map, and then it has a series of 1×1 convolutions to create question-specific bimodal embeddings. These embeddings are recurrently aggregated and then fed to a classifier that predicts the answer. Despite being composed of relatively simple elements, PReFIL outperforms more complex methods that use RNs and attention mechanisms. The three main stages of PReFIL are described in the next subsections. For DVQA, an additional fourth OCR-integration component is required (Sec. 3.4). In Sec. 4.3, we conduct studies to understand the value of each stage.

3.1. Multi-stage Image Encoder

For all model variants, image encoder is a DenseNet [12] trained from scratch. DenseNet is an efficient architecture for training deep convolutional neural networks (CNNs). It is comprised of several "dense blocks" and "transition blocks" between the dense blocks. Each dense block has several convolutional layers, where each layer uses outputs of all preceding layers as its input. The transition block sits between two dense blocks and serves to change feature-map sizes via convolution and pooling. This architecture encourages feature reuse, improves training, and mitigates vanishing-gradients, making it easy to train very deep networks. Feature reuse allows DenseNet to learn complex visual features with fewer parameters compared to other architectures [11].

In deep CNNs, complex features are learned as a hierarchy of visual features with earlier layers learning simple features and later layers learning higher-level features that are combinations of simpler features [41]. In data visualizations, simpler features such as color patches, lines, textures, etc. convey important information that is often abstracted away by deeper layers of a CNN. Hence, we use both low- and high-level convolutional features in our model, both of which are fed to parallel fusion module alongside question embeddings learned using an LSTM. We study the importance of both low and high level features in Sec. 4.3.

3.2. Parallel Fusion of Image and Language

Jointly modulating visual features using vision and language features can allow models to learn richer features for downstream tasks [29, 31, 37]. Our Q+I fusion block does this by first concatenating all of the input convolutional feature map's spatial locations with the question features, and then bimodal fusion occurs using a series of layers that use 1×1 convolutions [29, 37]. This allows the question to modulate visual feature processing and yields bimodal embeddings that capture information from both the image and the question. This approach resembles early VQA models that concatenated CNN embeddings to question embeddings, with the critical difference being that this happens

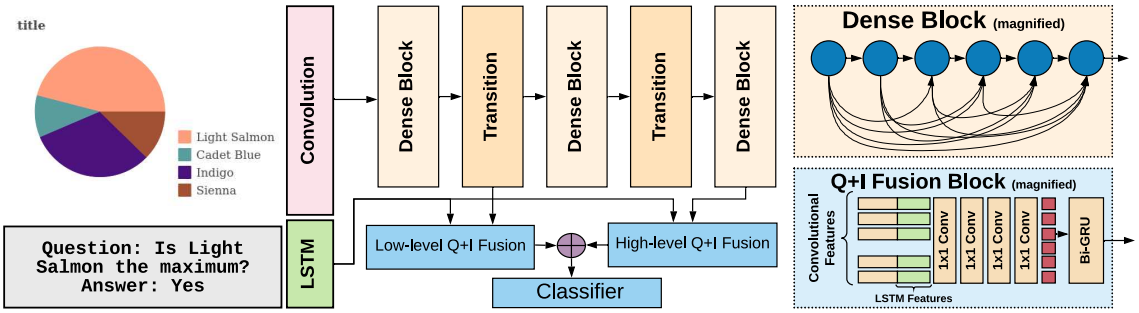


Figure 3. Components of our PReFIL model. Magnified views show the details of each dense block and Q+I fusion block.

before spatial pooling across the entire scene. We do this for both low-level and high-level convolutional features in parallel. In Sec. 4.3, we study the importance of learning bimodal embeddings jointly.

3.3. Recurrent Aggregation of bi-modal features

In CNNs, the most common approach to aggregating information from a feature map $F \in \mathbb{R}^{M \times N \times D}$ is to collapse across the spatial dimensions to produce a D dimensional vector by mean pooling or max pooling. An alternative is to "flatten" F to turn it into a DMN-dimensional vector. Recent attentive approaches have explored using a weighted sum, where the relative importance of each region is based on the question. These methods may fail to capture *interactions* among features, especially for high-level tasks such as question answering. To address this, we aggregate information using a bidirectional gated recurrent unit (bi-GRU), which sequentially takes in the D-dimensional features from each of the MN locations in F. The aggregated features are sent to a classifier to predict the answer. As ablation, we also try sum-pooling for aggregation in Sec. 4.3.

3.4. OCR Integration for DVQA dataset

Unlike FigureQA and most VQA tasks, DVQA requires OCR to answer its reasoning and data questions. A fixed vocabulary consisting of all the words seen during training is not enough since the model will encounter OOV words during testing. To integrate OCR into PReFIL, we use the same dynamic encoding scheme used by the SANDY model [16]. Dynamic encoding creates an image specific dictionary that associates the spatial positions of scene elements with entries in the dictionary. Before running the net, all words are detected using OCR and then they are associated with the appropriate element in the dynamic encoding dictionary based on each word's spatial position. Subsequently, if a question word is encountered that is in the dynamic dictionary then the appropriate element is set to 1. For answers, a portion of the classification layer is reserved for the dynamic encoding outputs. See [16] for additional details.

To assess impact of OCR, we test three OCR versions as well as a version of algorithm trained without the dynamic encoding, i.e., only using a fixed-vocabulary constructed from the train split. The first two OCR systems are identical to those used by [16]: an oracle (perfect) OCR model and a real OCR system using Tesseract. Because Tesseract has been found to be sub-optimal when used directly on diagrams [24], we also study using a two-stage OCR pipeline where we first detect text and then run OCR on the detected regions to recognize the text. Specifically, we use the EAST text detector [42] to detect text-regions for images rotated at 0, 45 and 90 degrees. We then perform non-maximum suppression on overlapping detections and crop them. Each cropped region is resized by 200% and sent to the Tesseract OCR to obtain the text within each region. The rest of the dynamic encoding scheme remains unchanged.

3.5. Model and Training Hyperparameters

Question Encoding: Question words are represented by 32 dimensional learned word embedding and passed through an LSTM which provides a 256-dimensional embedding representing the whole question.

DenseNet: We use a 40 layer DenseNet composed of 3 dense blocks with 12 layers each. The number of initial filters is 64 and the growth rate is set to 32.

Preprocessing: DVQA images are resized to a size of 256×256. FigureQA images are all differently sized but we resize them to 320×224 which maintains an *average* width-height aspect ratio. For data augmentation during training, both DVQA and FigureQA images are padded with 8 pixels on all sides, followed by random crops and random rotations of up to 3 degrees.

Q+I Fusion: Inputs to Q+I block are batchnormed. Each Q+I fusion block is composed of four 1×1 convolutions with 256 channels and ReLU.

Recurrent Fusion: The bimodal features are aggregated using a 256 dimensional bi-directional GRU. The forward and backward direction outputs are concatenated to form a 512 dimensional vector which is fed to the classifier.

Classifier: The aggregated bimodal features are projected to a 1024 fully connected ReLU layer, which was regularized using dropout of 0.5 during training. The classification layer is binary for FigureQA. For DVQA, the clas-

sification layer has 107 units, with 77 units for predicting 'common' answers such as 'yes', 'no', 'three groups', etc, and 30 special tokens for predicting answers that require OCR, which allows PReFIL to produce OOV answer tokens that are unseen during training (see Sec. 3.4 for details).

Losses and Optimizers: For DVQA, PReFIL is trained using multinomial cross-entropy loss. For FigureQA, PReFIL is trained using binary cross entropy loss. Following [26], we use Adamax optimizer with a gradual learning rate warm-up, with a base learning rate of 7×10^{-4}. The first 4 epochs use a learning rate of $(0.5 \times epoch \times base)$ and the rate starts decaying by a factor of 0.7 from epochs 15 to 25. For DVQA, all models are trained for a fixed 25 epochs. For FigureQA, we train them until they converge on the validation set and submit predictions to its creators for assessment on the non-public test set.

4. Experiments and Results

4.1. FigureQA

FigureQA has two validation sets and two non-publicly available test sets. Validation 1 and Test 1 have the same colors as the training set and Validation 2 and Test 2 have a color scheme that differs from training. Test sets are not publicly available and the results were obtained by sending the predictions to the authors. Existing works do not report accuracy for the full test set, but we report results for both validation and test sets in Table 3.5 for completeness.

Our PReFIL algorithm exceeds FigureNet by a large margin despite FigureNet having access to additional annotations. FigureNet is incapable of answering questions about line and dot-line graphs, so it is only evaluated on vBar, hBar and Pie. For these chart types, average accuracy for FigureNet is 83.9%, compared to 97.33% for ours.

FigureQA also provides human performance for a *subset* of Test 2, which is not available for the other sets. We report PReFIL's performance compared to other baselines and human performance on the exact same subset in Table 3. PReFIL outperforms the human baseline for four out of five categories and also surpasses overall human accuracy. When analyzed for different question templates, PReFIL outperforms humans for 12 out of 15 question templates. PReFIL shows the most improvements for questions requiring measurements, e.g. for the question template "Is X the high/low median?" PReFIL outperforms human accuracy by over 7% (absolute). Detailed results for all 15 templates are presented in the supplementary materials.

4.2. DVQA

DVQA is split into Test-Familiar, which contains bar charts with words that are also encountered in its Train set, and Test-Novel, which contains bar charts with novel words in them. Results for both DVQA splits are given in Table 4. PReFIL surpasses SANDY by over 40% in accuracy when both the baseline SANDY and our PReFIL method have access to a perfect Oracle OCR, which is emulated by providing the correct text-annotations for all the elements in the images. When using Tesseract OCR, we obtain about a 24% improvement overall on both test sets. To demonstrate that PReFIL's performance scales with access to better OCR, we also test a version that uses an improved OCR pipeline (see Sec. 3). This further improves PReFIL's performance by about 11% bringing it closer to the results of the oracle OCR version. When OCR is removed entirely, PReFIL still performs about 11% better than SANDY without OCR, but this ablation renders many data and reasoning questions impossible to answer. This re-affirms the assertion by DVQA's creators that OCR integration is essential for answering the data and reasoning questions in the dataset [16].

Across all OCR variants, PReFIL outperforms SANDY. Moreover, PReFIL's performance scales much better when better OCR is available: 11% gain for SANDY vs. 26% gain for PReFIL when moving from the imperfect Tesseract OCR setup to the perfect Oracle OCR setup. Our results show that PReFIL is as effective for novel words (Test-Novel) as it is for familiar words (Test-Familiar). This is enabled by the dynamic OCR integration, which is designed to be agnostic to whether a word has been encountered before.

Because no human accuracy estimate for DVQA existed, we had people answer 5000 randomly selected questions for 5000 images from the DVQA Test-Novel split. The annotators were shown example QA pairs from each of three question types. We perform post processing on the provided answers to rectify minor answer entry errors. First, we found some annotators used decimal points or spelled out numerals ("5.0" or "five" instead of "5") despite our instructions to only use integers when answers are numbers. Because DVQA contains only integers, we convert all such occurrences to the nearest integer. For word answers, we allow one character typographic error to be discounted. Results for humans and models are given in Table 4. With perfect OCR, PReFIL surpasses the DVQA human accuracy result across question types. Its performance on reasoning questions is almost 10% greater (absolute), and it exceeds them by almost 8% (absolute) for DVQA's data questions, which require measurement. However, without perfect OCR humans exceed PReFIL, although the better OCR used for PReFIL does lead to significantly better results than PReFIL with improved OCR. This suggests that the underlying core algorithm and reasoning mechanisms in PReFIL work well for DVQA, and the main limiting factor is OCR.

4.3. Ablation Studies

We studied the contribution of PReFIL's components by analyzing a series of ablation models. We trained each model variation and the original PReFIL (Oracle OCR) for

Table 2. Results for the FigureQA dataset for our PReFIL algorithm compared to baseline and existing algorithms.

	Validation 1 - Same Colors						Validation 2 - Alternated Colors					
	vBar	hBar	Pie	Line	Dot-line	Overall	vBar	hBar	Pie	Line	Dot-line	Overall
QUES [21]	-	-	-	-	-	-	-	-	-	-	-	50.01
IMG+QUES [21]	61.98	62.44	59.63	57.07	57.35	59.41	58.60	58.05	55.97	56.37	56.97	57.14
RN [21]	85.71	80.60	82.56	69.53	68.51	76.39	77.35	77.00	74.16	67.90	69.04	72.54
FigureNet [33]	87.36	81.57	83.13	-	-	-	-	-	-	-	-	-
PReFIL (Ours)	**98.80**	**98.09**	**95.11**	**91.82**	**92.19**	**94.84**	**98.46**	**97.94**	**93.57**	88.50	**90.30**	**93.26**
	Test 1 - Same Colors						**Test 2 - Alternated Colors**					
PReFIL (Ours)	98.79	98.14	95.35	91.98	92.05	94.88	98.41	97.93	93.58	88.26	90.07	93.16

Table 3. Results on FigureQA's Test 2 split with alternated color schemes. All results are from the 16,876 questions answered by human annotators.

Type	PReFIL(Ours)	Q+I [21]	RN [21]	Human [21]
vBar	**98.25**	59.63	77.13	95.90
hBar	**97.98**	57.69	77.02	96.03
Pie	**92.84**	55.32	73.26	88.26
Line	87.79	54.46	66.69	**90.55**
Dot-line	**89.57**	54.19	69.22	87.20
Overall	**92.79**	56.04	72.18	91.21

25 epochs on a subset of DVQA that has only 500,000 randomly selected training samples. The ablation models are:

- **No bimodal embeddings:** Instead of learning bimodal embeddings, the question is concatenated after the recurrent aggregation and fed to the classifier.
- **No low-level features:** Only the high-level (layer 40 output) DenseNet features are used.
- **No high-level features:** Only the low-level (layer 14 output) DenseNet features are used. This is equivalent to using a shallower DenseNet.
- **No recurrent aggregation:** Instead of recurrent aggregation, output is aggregated via summation.

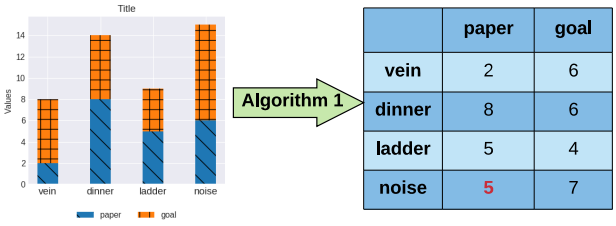
As shown in Table 5, all of PReFIL's components impact its performance. Removing bimodal embeddings causes the largest accuracy drop (over 12% absolute). The next largest is caused by removing low and high-level visual features (1.3% and 6% absolute).

4.4. Table Reconstruction by Asking Questions

We introduce table reconstruction for DVQA as an application of PReFIL. DVQA's question templates provide the questions needed to completely reconstruct its bar charts by iteratively asking questions about each chart. Our approach is given in Algorithm 1. An example reconstruction is shown in Fig. 4, and results using PReFIL (Oracle OCR) are given in Table 6. Shape prediction can be done with near perfect accuracy, but there is a drop in performance for both label and value prediction. To study the accuracy of different components in chart reconstruction, we also report accuracy on three main components of the iterative question-answering: 1) Shape prediction: Questions about number of bars and legends in the picture; 2) Label prediction: Predicting the label of given bar or legend; and 3) Value Prediction: Predicting the value of a given bar.

Algorithm 1: Iterative QA for Data Reconstruction

if *bar_type is single* **then**
  n = ans("How many bars are there?");
  for $i \leftarrow 1$ *to* n **do**
    data[i] = ans("What is the value of the i^{th} bar?");
    label[i] = ans("What is the label of the i^{th} bar?");
else
  m = ans("How many groups are there?");
  n = ans("How many bars are there per group?");
  for $j \leftarrow 1$ *to* n **do**
    legend_label[j] = ans("What is the label of the j^{th} bar in each group?");
  for $i \leftarrow 1$ *to* m **do**
    bar_label[i] = ans("What is the label of the i^{th} group?");
    for $j \leftarrow 1$ *to* n **do**
      data[i,j] = ans("What is the value of the j^{th} bar in i^{th} group?");



	paper	goal
vein	2	6
dinner	8	6
ladder	5	4
noise	5	7

Figure 4. An example output of the chart to table algorithm. Red denotes incorrect predictions.

Table 4. Results for the DVQA dataset for PReFIL compared to baselines and existing algorithms.

	Test-Familiar				Test-Novel			
	Structure	Data	Reasoning	Overall	Structure	Data	Reasoning	Overall
QUES [16]	44.03	9.82	25.87	21.06	43.90	9.80	25.76	21.00
IMG+QUES [16]	90.38	15.74	31.95	32.01	90.06	15.85	31.84	32.01
SANDY (No OCR) [16]	94.71	18.78	37.29	36.02	94.82	18.92	37.25	36.14
PReFIL (No OCR)	99.77	23.39	49.05	47.70	99.77	23.43	49.21	47.86
SANDY (Tesseract OCR) [16]	96.47	37.82	41.50	45.77	96.42	37.78	41.49	45.81
PReFIL (Ours, Tesseract OCR)	99.75	49.00	74.61	69.63	99.73	48.91	74.07	69.53
PReFIL (Ours, Improved OCR)	99.73	68.55	83.44	80.88	99.57	67.13	80.73	80.04
SANDY (Oracle OCR) [16]	96.47	65.40	44.03	56.48	96.42	65.55	44.09	56.62
PReFIL (Ours, Oracle OCR)	**99.77**	**95.80**	**95.86**	**96.37**	**99.78**	**96.07**	**95.99**	**96.53**
Human	-	-	-	-	96.19	88.70	85.83	88.18

Table 5. PReFIL ablation studies on a 500K DVQA train subset.

Ablation Model	Test Familiar	Test Novel
PReFIL (full model)	**91.18**	**91.32**
No bimodal embedding	78.00	78.36
No high-level features	85.68	85.86
No low-level features	89.87	90.05
No recurrent aggregation	90.88	91.14

Table 6. Bar chart reconstruction accuracy (%) using Algorithm 1 with PreFIL (Oracle OCR).

	Test Familiar	Test Novel
Shape Prediction	99.97	99.97
Label Prediction	97.78	97.78
Value Prediction	84.21	84.75
Overall	90.79	91.10

5. Discussion

PReFIL surpassed prior state-of-the-art methods for both DVQA and FigureQA. While PReFIL exceeded the human baseline for FigureQA, results are more nuanced for DVQA due to OCR model variations. All OCR versions exceeded the human baseline for structure questions, but only PReFIL using oracle OCR exceeded humans across all question types. We found that better OCR methods led to better results for DVQA. Future developments in OCR technology would likely improve PReFIL further.

The strong results in this paper suggest that the community is ready for more difficult CQA datasets. We have the following recommendations:

- **Charts in the wild:** The charts in FigureQA and DVQA were methodologically generated, but human-generated charts in real-world business and scientific documents can contain variations that these datasets omit. Additional text in the chart or human annotations would likely cause the dynamic encoding method used by PReFIL to fail. Next generation datasets should contain charts extracted from real-world documents.
- **Human generated questions:** The questions in both FigureQA and DVQA were created with templates, which do not capture all the nuances of natural language. Deploying a chart question answering system will require it to handle human-generated queries. Studies on the synthetically generated CLEVR dataset have demonstrated that algorithms experience a large drop in performance when natural language questions are asked to a model trained only on CLEVR [15]. Future CQA datasets should include human-generated question-answer pairs.
- **Document-level CQA:** FigureQA and DVQA have well-defined image regions and all information needed to answer a question is contained in that image. To understand charts in documents, information in the rest of the document may be necessary to answer questions about the chart. Beyond typical CQA algorithm abilities, this requires document question answering [8], page segmentation [10], and more. Creation of such a dataset would greatly increase the challenge for future algorithms and better match real-world usage.

6. Conclusion

We proposed PReFIL, a new CQA system that improves the state-of-the-art and surpasses human accuracy on two datasets. Like other VQA tasks [19], our results suggest harder datasets are needed. For CQA, better OCR is also important for advancing the field. Our work has the potential to improve retrieval of information from charts, which has numerous applications, including automatic information retrieval, table reconstruction, and enabling better understanding of charts by people with visual impairments.

Acknowledgements. This research was supported in part by NSF award #1909696 to C.K. We thank NVIDIA for gifting a GPU to C.K.'s lab.

References

[1] M. Acharya, K. Jariwala, and C. Kanan. VQD: Visual query detection in natural scenes. In *NAACL*, 2019. 2

[2] M. Acharya, K. Kafle, and C. Kanan. TallyQA: Answering complex counting questions. In *AAAI*, 2019. 1, 2

[3] A. Agrawal, D. Batra, and D. Parikh. Analyzing the behavior of visual question answering models. In *EMNLP*, 2016. 2

[4] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, 2018. 2

[5] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 4

[6] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Deep compositional question answering with neural module networks. In *CVPR*, 2016. 2

[7] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual question answering. In *ICCV*, 2015. 1, 2

[8] C. Clark and M. Gardner. Simple and effective multi-paragraph reading comprehension. In *ACL*, 2018. 8

[9] M. Cliche, D. Rosenberg, D. Madeka, and C. Yee. Scatteract: Automated extraction of data from scatter plots. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017. 2

[10] D. He, S. Cohen, B. Price, D. Kifer, and C. L. Giles. Multi-scale multi-task fcn for semantic page segmentation and table detection. In *ICDAR*, 2017. 8

[11] G. Huang, S. Liu, L. van der Maaten, and K. Q. Weinberger. CondenseNet: An efficient densenet using learned group convolutions. In *CVPR*. 2018. 4

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 4

[13] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017. 1

[14] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 1, 2

[15] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. Inferring and executing programs for visual reasoning. In *ICCV*. 8

[16] K. Kafle, S. Cohen, B. Price, and C. Kanan. Dvqa: Understanding data visualizations via question answering. In *CVPR*, 2018. 1, 2, 4, 5, 6, 8

[17] K. Kafle and C. Kanan. An analysis of visual question answering algorithms. In *ICCV*, 2017. 2

[18] K. Kafle and C. Kanan. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 2017. 1, 2

[19] K. Kafle, R. Shrestha, and C. Kanan. Challenges and prospects in vision and language research. *Frontiers in Artificial Intelligence*, 2019. 2, 8

[20] K. Kafle, M. Yousefhussien, and C. Kanan. Data augmentation for visual question answering. In *INLG*, 2017. 1

[21] S. E. Kahou, A. Atkinson, V. Michalski, A. Kadar, A. Trischler, and Y. Bengio. FigureQA: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*, 2017. 1, 2, 7

[22] J. S. Kallimani, K. Srinivasa, and R. B. Eswara. Extraction and interpretation of charts in technical documents. In *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, pages 382–387. IEEE, 2013. 2

[23] V. Kazemi and A. Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*, 2017. 4

[24] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016. 2, 5

[25] A. Kembhavi, M. Seo, D. Schwenk, J. Choi, A. Farhadi, and H. Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *CVPR*, 2017. 2

[26] J.-H. Kim, J. Jun, and B.-T. Zhang. Bilinear attention networks. In *NeurIPS*, 2018. 6

[27] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 2

[28] J. J. Lau, S. Gayen, A. B. Abacha, and D. Demner-Fushman. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5:180251, 2018. 1

[29] M. Malinowski and C. Doersch. The visual QA devil in the details: The impact of early fusion and batch norm on clevr. *arXiv preprint arXiv:1809.04482*, 2018. 4

[30] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NeurIPS*, 2014. 1, 2

[31] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *AAAI*, 2018. 4

[32] J. Poco and J. Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. In *Computer Graphics Forum*, volume 36, pages 353–363. Wiley Online Library, 2017. 2

[33] R. Reddy, R. Ramesh, A. Deshpande, and M. M. Khapra. A question-answering framework for plots using deep learning. *arXiv preprint arXiv:1806.04655*, 2018. 4, 7

[34] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *NeurIPS*, 2015. 2

[35] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *NeurIPS*, 2017. 4

[36] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th annual*

ACM symposium on User interface software and technology, pages 393–402. ACM, 2011. 2

[37] R. Shrestha, K. Kafle, and C. Kanan. Answer them all! toward universal visual question answering models. In *CVPR*, 2019. 4

[38] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach. Towards vqa models that can read. In *CVPR*, 2019. 1

[39] S. Tsutsui and D. J. Crandall. A data driven approach for compound figure separation using convolutional neural networks. In *ICDAR*, volume 1, pages 533–540. IEEE, 2017. 2

[40] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 4

[41] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015. 4

[42] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: an efficient and accurate scene text detector. In *CVPR*, 2017. 5

[43] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016. 2