CapNet: Making Science More Accessible via a Neural Caption Generator

Stanford CS224N Custom Project

Aman Lada
Department of Computer Science
Stanford University
amanl@stanford.edu

Tirth Surti
Department of Computer Science
Stanford University
tsurti@stanford.edu

Abstract

Graphs, charts and other figures are an important component of scientific discourse but present a formidable challenge to the visually impaired. While formal publications include captions, these are generally missing in informal scientific media like blogs and discussion forums. We present a neural approach to this problem, allowing quick and accurate generation of captions for unlabeled figures. We experiment with two types of architectures - a baseline CNN+transformer approach, and an experimental one that utilizes vision transformers (ViTs) and pretrained language models (LMs). These models are trained on $N = 20,000$ image-caption pairs drawn from a larger corpus of $620,000$ data points. By comparing BLEU-4 and ROUGE-4 scores across different encoder-decoder combinations, we find that a ViT with large input dimensions (384×384) coupled with a GPT-2 LM beats the state of the art by a wide margin (BLEU-4 of 6.58 and ROUGE-4 F-1 of 0.044). We also perform qualitative analysis through a single blind study, suggesting that such an approach is capable enough to handle complex graphs but suffers from considerable hallucination. Further exploration with multi-modal encoders that incorporate visual and text (OCR) inputs is likely to mitigate some of these issues.

1 Key Information

Mentor: David Lim
Team Contributions: Both team members researched and devised the experimental set-up together. Tirth Surti created the baseline code and wrote the 'Related Works' & 'Approaches' section of the report. Aman Ladia worked on the Vision Encoder Decoder code and wrote the 'Experiments' and 'Analysis' sections. 'Abstract' and 'Conclusion' was written together.

2 Introduction

Problem. Graphical figures like charts and graphs are an integral part of scientific literature (Shin et al., 2001). While formal texts like research papers tend to include captions underneath such diagrams, informal channels like blogs, discussion boards and lecture notes rarely do. Even when captions exist, they tend to be poorly written and nondescript (Franzblau and Chung, 2012). This poses a significant accessibility challenge as the visually impaired community struggles to partake in scientific discourse. The goal of this paper is to mitigate this issue through a neural captioning system that is specifically designed for scientific diagrams. This enables blind individuals to interpret captionless figures with ease.

Existing Methods. While this problem falls under the broader banner of image captioning, it has unique requirements that warrant a specialized approach. Unlike generalized captioning tasks where feature extractors can rely on prominent entities, shapes and colors in the input, these are generally

moot with scientific figures. Instead, characteristics like the legend, axis labels and position of lines/bars are much more significant. Moreover, caption length is key - it must be sufficiently long to capture key details but also concise enough not to regurgitate axis values. As a result, general image captioning models do not perform well on this task. Past work in this space is sparse; the approaches that do exist are outdated, using LSTM based systems that perform better than general models but do not produce very cohesive captions (Vinyals et al., 2014). These are discussed further in section 3.

Overview of Approach. To overcome these limitations, we propose two approaches: a baseline model that replicates prior work and uses transformers rather than LSTMs, and a vision encoder decoder (VED) model that utilizes vision transformers (ViTs) with pre-trained decoders to significantly improve output captions. These models were trained on over $20,000$ image-caption pairs drawn from arXiv papers (see section 5.1). We found that the VED models exceed the state of the art. This results includes performance on both quantitative measures (BLEU 1-4 and ROUGE) as well as qualitative evaluation through a single-blind study.

3 Related Work

Our work to caption scientific images falls along the broad category of image caption generation. Earlier methods of image caption generation have used a combination of CNN-based encoders and LSTM-based decoders (Vinyals et al., 2014; Xu et al., 2016; Anderson et al., 2018). Since then, improvements have been made on both the encoder and decoder ends for improved feature extraction from images and sequence generation from decoders. In particular, we improve upon these methods in our baseline, using a transformer-based decoder instead. With their multi-headed self-attention mechanism, transformers avoid the necessity of hard-to-train recurrent layers and have been shown to outperform recurrent neural networks, particularly for textual generation (Vaswani et al., 2023).

Rather than using a CNN-based encoder and LSTM-based decoder, Wang et al. (2022) use a complete transformer-based encoder and decoder for image captioning. These authors use the Swin Transformer (Liu et al., 2021) to extract image features by partitioning the input image into patches which are treated like tokens of a sentence. It then iteratively merges these patches to produce a hierarchical representation of features. Such an approach is less computationally expensive than standard CNNs and is better at extracting higher-level features rather than local features. This is important for scientific image processing as captions should represent global information of trends in the figure. Given the global features extracted from the image, the authors then use a transformer-based decoder with multi-headed self-attention for sequence generation.

Approaches to evaluating the quality of the generated captions have remained consistent throughout these different implementations. In particular, BLEU scores using 1-grams to 4-grams have frequently been used (Papineni et al., 2002). Since BLEU scores check for exact matching of n-grams and therefore may not be a robust method for checking similarity between captions, some approaches also use ROUGE (Lin, 2004a). This metric further emphasizes sentence structures and coherent captions (through sentence subsequence matching), which we seek to use in our work. Ultimately, on the MSCOCO object-detection dataset, Vinyals et al. (2014) obtained a BLEU-4 score of 27.7 while Wang et al. (2022) obtain a significantly higher BLEU-4 score of 41.4, revealing the power of transformer based encoders and decoders for general image captioning.

4 Approach

Our approach involves exploring two different models for image captioning: a baseline ResNet-50 CNN based encoder and transformer based decoder and multiple pretrained vision encoder decoder (VED) models, which we have coded on our own.

4.1 Data Preprocessing

4.1.1 Baseline Model

Our dataset consists of N RGB images $\mathbf{X}^{(i)} \in \mathbb{Z}^{n \times m \times 3}$ over variable dimensions $n \times m$. In order to use the images, we must rescale to a fixed size and normalize appropriately. Without loss of generality, assuming that $n < m$, we rescale the image so that $n = 256$ and then crop the image at the center

so that $\mathbf{X}^{(i)} \in \mathbb{Z}^{224\times224\times3}$ across the dataset. We then normalize the images across channels with means $\mu = (0.485, 0.456, 0.406)$ and $\sigma = (0.229, 0.224, 0.225)$ as done in the ResNet-50 model (He et al., 2015). In particular, for a given channel in our image $\mathbf{X}_c^{(i)} \in \mathbb{Z}^{224\times224}$, for $c \in \{1, 2, 3\}$, we normalize each channel by taking:

$$\mathbf{X}_c^{(i)\prime} = \frac{\mathbf{X}_c^{(i)} - \mu_c}{\sigma_c} \in \mathbb{R}^{224\times224}. \tag{1}$$

In order to pre-process the captions, we need to develop a vocabulary among the corpus that we have. In order to construct a vocabulary, we first generate a preliminary vocabulary of size $|V|$, tokenizing every caption in the dataset and finding all unique words. We do so by stripping any punctuation, fixing lowercase, and including the necessary tokens for padding, start/end indicators, and unknown tokens: '<pad>', '<start>', '<end>', '<unk>', respectively. In order to speed up training we use the vocabulary of the pre-trained GloVe embeddings Pennington et al. (2014) with dimension $e = 100$ and vocabulary size of $\sim 400,000$. We then update the vocabulary with only words that appear in the GloVe vocabulary so that we access to their associated embedding. We fix the zero vector to be the padding token's embedding: $\mathbf{0} \in \mathbb{R}^e$. For the start/end/unknown tokens, we generate a uniform random vector $\mathbf{x} \in \mathbb{R}^e$, where $x_i \sim \text{Uniform}(-1, 1)$ for the embedding. We also fix captions to a maximum length of 256, adding necessary pad tokens after the end-of-sentence token.

4.1.2 VED Models

Here, we utilize the pretrained ViT (Vision Transformer) feature extractor to generate an input sequence for the VED models (Dosovitskiy et al., 2021). We use ViT to extract features from rescaled images of size $(224, 224, 3)$ as well as $(384, 384, 3)$ in order to capture more details. These images are normalized for each channel with means $\mu = (0.5, 0.5, 0.5)$ and standard deviations $\sigma = (0.5, 0.5, 0.5)$. Assuming a rescaled square input image of size n, the feature extractor works by converting images $\mathbf{X}^{(i)} \in \mathbb{R}^{n\times n\times3}$ into a sequence of m non-overlapping patches of size $\mathbf{X}_p^{(i)} \in \mathbb{R}^{m\times3p^2}$, where p^2 is the number of pixels in the patch and $m = n^2/p^2$. Note that for input images of size $(224, 224, 3)$, $p = 16$ where as for input images of size $(384, 384, 3)$, $p = 32$. Similar to the baseline, these patches are projected into a pre-trained embedding space with added positional encodings to preserve the relative spatial ordering of the patches.

For the captions, instead of explicitly generating a vocabulary to map captions to index tensors, we utilize various pre-trained tokenizers: BERT, RoBERTa, and GPT-2 (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019b). All of these tokenizers use iterative merging techniques to create a vocabulary of subwords and characters that appear with the highest frequencies. This enables these tokenizers to be robust against the technical jargon that will often be seen in our caption dataset. We fix the same maximum caption length as the baseline.

4.2 Models

4.2.1 Baseline Model

For the baseline model (see Figure 1), we use a pretrained ResNet-50 model to extract features from the images and generate a feature sequence for input. We then pass the feature-extracted image and its caption into a transformer-based decoder to generate predictions for the output sequence.

The ResNet-50 architecture consists of 16 residual layers, each involving multiple convolutional layers with ReLU activation for robust feature extraction (He et al., 2015). Since the original architecture is intended for classification, we remove the last two layers, so that the final layer returns the feature map from the final residual layer. In order to generate a sequence from the features, we flatten the 2D feature map and project down to the desired hidden size of 512. Hence, an input image of size $(224, 224, 3)$ will get mapped to a sequence of size $(512, 49)$, where 49 is the sequence length. Note that we freeze all of the layers of the Resnet-50 model apart from the final projection layer. This allows for some minimal finetuning of the features before being passed into the decoder.

For the decoder, we implement the transformer architecture of Vaswani et al. (2023), using the standard 6 decoder layers, 8 attention heads, and a feedforward dimension of 2048. We first retrieve the input captions' GloVe embeddings and project them to a target size of $(256, 512)$, where 256 is the maximum (padded) sequence length. We then pass the sequence through a fixed positional

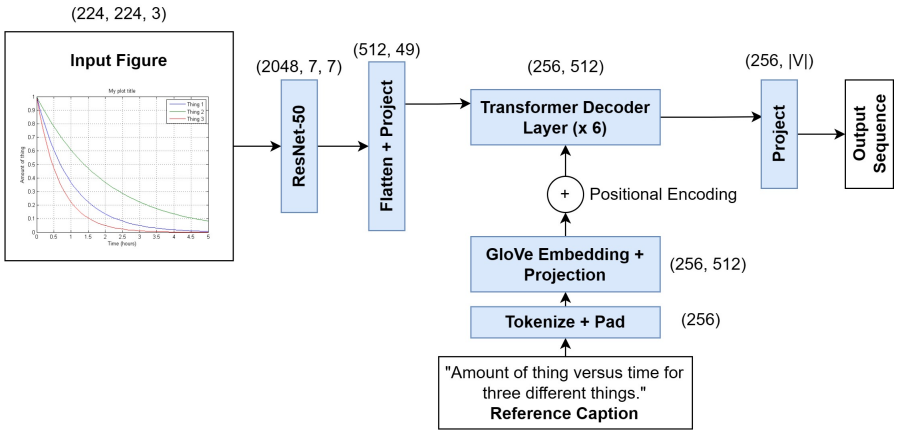


Figure 1: Workflow of the baseline model. The shapes labeled for each block are the output sizes of that block.

encoder in order to model the relative positions of each token in a sequence. In particular, for each position ℓ in the target sequence and feature index i, we add a positional encoding PE at index (ℓ, i) into the projected embeddings where

$$PE(\ell, i, x) = \begin{cases} \sin\left(\frac{\ell}{x^{2i/\ell}}\right), & i \equiv 0 \pmod 2 \\ \cos\left(\frac{\ell}{x^{2i/\ell}}\right), & i \equiv 1 \pmod 2 \end{cases}.$$

We choose $x = 10\,000$, which is large enough to provide a unique addend to each position in the projected feature tensor of the captions and therefore enables the decoder to better understand the relative order of tokens in the captions. We incorporate a mask for each input caption so that the padded tokens are ignored during attention calculations. To get logits for caption decoding and loss calculations, we pass the output of the transformer decoder through a linear layer to project it to the vocabulary size $|V|$, giving an output tensor O of size $(256, |V|)$. In order to generate the decoded caption for evaluation purposes, we utilize a greedy approach, taking the vocabulary word with the maximum logit for every sequence position: $\max_j |O_{ij}|$.

4.2.2 VED Models

For the VED models (see Figure 2), since we are using ViT feature extractor, we correspondingly use the base version of the ViT transformer-based encoder, which converts the sequence of the flattened and positionally encoded image patches into a context vector that can be passed into a decoder (Dosovitskiy et al., 2021).

Similarly, as we have used the BERT, RoBERTa, and GPT-2 tokenizers, we use their corresponding pre-trained transformer-based decoders for the output sequence generation (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019b). We seek to compare the effectiveness of these decoders as they have been trained on different corpuses and have different number of parameters for finetuning. In particular, while BERT and RoBERTa were trained on the same corpus (Wikipedia and BookCorpus), the base RoBERTa model further optimizes the potentially undertrained BERT and allows for more parameters. GPT-2 on the other hand has far more parameters than the other two and was optimized for textual generation over a different dataset of web-scraped sources, which will be useful for our scientific caption generator. Captions are decoded in the same way as the baseline.

4.3 Loss Function

The decoders of the models we use output an (unnormalized) probability distribution over the vocabulary for each position in the output sequence. Naturally, we want to optimize over how close this distribution is to that of the original input caption. We therefore optimize our models over the cross-entropy loss. In particular, let $L = 256$ be the length of the sequence and $|V|$ be the size of the vocabulary. Define V_i to be the index of the word in the vocabulary of the model we are using for the

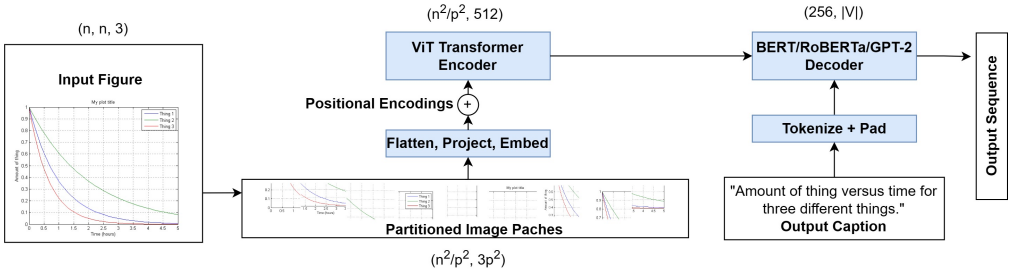


Figure 2: Workflow of the VED Models, including image patching done by ViT and the tokenization done by BERT, RoBERTa, and GPT-2.

word at position i in the input caption. Then, for each logit $o_i \in \mathbb{R}^{|V|}$ in the output sequence, we compute the cross entropy as:

$$\ell_i(V_i, o_i) = -\log \frac{\exp(o_i[V_i])}{\sum_{j=1}^{|V|} \exp(o_i[V_j])} \mathbb{1}[V_i \neq V[\text{`<pad>'}]]. \tag{2}$$

Note that we ignore the contribution of padding tokens in the input caption to the cross entropy loss with the indicator function. We then sum over the entire sequence to get the mean loss:

$$L(\mathbf{o}, \mathbf{V}) = \frac{1}{\sum_{i=1}^{L} \mathbb{1}[V_i \neq V[\text{`<pad>'}]]} \sum_{i=1}^{L} \ell_i(V_i, o_i). \tag{3}$$

We optimize this loss using Adam (Kingma and Ba, 2017).

5 Experiments

5.1 Data

The starting point for our data was the SciCap Scientific Figures Dataset by Hsu et al. (2021), which contains over 416,000 figures and captions from computer science arXiv papers uploaded between 2010 and 2020. We also scraped 200,000 figures ourselves from other arXiv categories like Physics and Quantitative Biology using the PDFFigures 2.0 tool by Clark and Divvala (2016). Although we used an NVIDIA Tesla T4 graphics card on a cloud compute instance to train our models, we found that a single epoch through all 620,000 figures would take ≈ 8 hours. Given limitations on time and compute resources, we sampled $N = 20,000$ images from the larger dataset to use as training data. Similarly, we sampled 4000 images each for our validation and test sets.

All images were pre-processed through re-scaling and captions with token replacement, as discussed in section 4.1. An example image/caption pair from our dataset is presented in figure 4.

5.2 Evaluation method

Quantitative Evaluation. We evaluated our model's scientific figure captions against the gold labels using BLEU-1, BLEU-2, BLEU-3 and BLEU-4 scores (Brownlee, 2017). The four BLEU values signal different characteristics about our captions: BLEU-1, for instance, signals how well our candidate captions are capturing basic words, while BLEU-4 captures longer phrases and clauses. Past work in this space has also traditionally used BLEU scores, making it easy to compare results.

Besides BLEU, we also used ROUGE-1,2,3 and 4 to evaluate model performance Lin (2004b). While BLEU measures precision (i.e. how many of the machine generated n-grams appear in human captions), ROUGE also measures recall (i.e. how many of the human n-grams appear in machine generated captions). This makes the two measures complimentary; for completeness, we compute the ROUGE-n F1-score, which combines the ROUGE precision (P) and recall (R) by computing

$$F1 = \frac{2PR}{P + R}$$

Qualitative Evaluation. Despite their merits, BLEU and Rouge scores are not robust to similarities in meaning that appear through synonyms; as a result, we also qualitatively evaluate the final captions generated by our models. This was done via a single blind study, where five independent colleagues were presented 82 figures with four accompanying captions. They were asked to rate each caption on a scale of $1 - 10$. These captions were drawn from the gold labels and the outputs of our models. We analyzed the scores given by our colleagues across different types of diagrams to evaluate the workings of our models along with their key successes and failures.

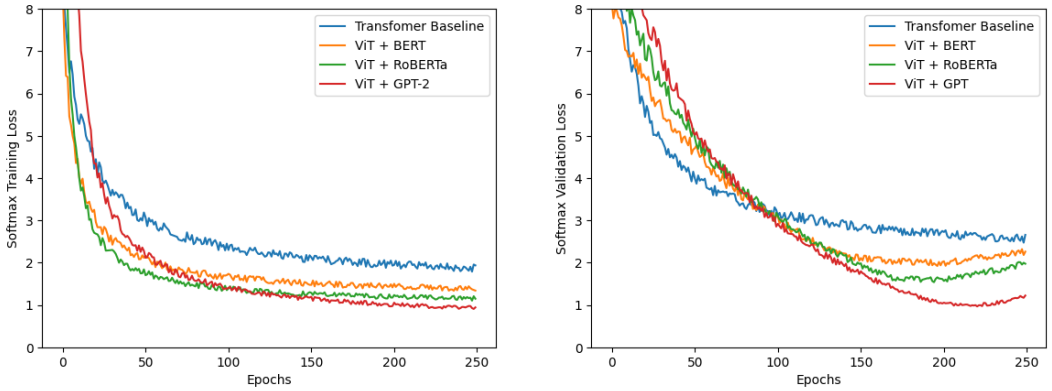
5.3 Experimental details

Given our large models and limited time, we sampled $N = 20,000$ images and captions from our training corpus to use as our trial training data. We set the maximum caption size to 256 tokens and an initial learning rate of 8×10^{-6}. The vocabulary size differed for each model: our baseline had $|V| = 12,144$, BERT had $|V| = 30,522$, RoBERTa had $|V| = 50,265$ and GPT-2 had $|V| = 50,257$. We also had 2800 warm-up steps; this allows the Adam optimizer to compute gradient statistics and enables the model to explore the loss landscape before narrowing the search space (Gotmare et al., 2018). Using a batch size of 32, which is standard among encoder-decoder routines, we trained each model over one full day ($\approx$ 24 hours) for 250 epochs. Given the large model size, the training was performed on a cloud compute instance with an NVIDIA Tesla T4 GPU.

5.4 Results

We begin by comparing the results for four key models: the transformer baseline, ViT with BERT, ViT with RoBERTa and ViT with GPT-2. For the latter three, we deployed a Vision Transformer encoder pre-trained on ImageNet-21k at a resolution of 224×224.

Loss Curves. Analysing the training loss in figure 3a, we see that all four models experience a sharp decline in the first 50 epochs. However, it is important to note that initially, the ViT + GPT-2 model performs worse than the three other models. By epoch 50, its loss drops below that of the baseline (≈ 2.5 vs ≈ 3.1), but continues to stay above BERT and RoBERTa (≈ 2.2 and ≈ 1.9 respectively). In fact, it is not until epoch 120 that GPT performs better than RoBERTa. At first this is counter-intuitive given that GPT-2 generally performs better than RoBERTa at language tasks; however, we explain this trend by analysing the number of parameters in each of these models. While BERT has 108 million parameters and RoBERTA has 123 million (Juan, 2021), GPT-2 is an order of magnitude larger at 1.5 billion parameters (Radford et al., 2019a). Given the highly niche task of scientific figure captioning which involves producing esoteric, domain-specific jargon, the larger number of parameters likely increase the number of epochs taken to fine tune GPT-2 (Wallace, 2020). After 250 epochs of training, the final training losses were 1.87, 1.395, 1.158 and 0.931 for the baseline, BERT, RoBERTa and GPT-2 models respectively.



(a) Training loss across 250 epochs.

(b) Validation loss across 250 epochs.

Figure 3: Comparing losses for baseline, ViT + BERT, ViT + RoBERTa and ViT + GPT-2 models

We now evaluate the cross-entropy loss on the validation set as shown in figure 3b. Initially, the trend was as expected - a sharp decline as the four models learnt the nature of the data. The baseline model's loss curve continued in a downward trajectory, plateauing at ≈ 3.1 around epoch 200. This suggests that the model does not have enough expressivity to capture variations in input data. However, the results for the VED models were starkly different. Although their validation losses declined initially, the curves took a parabolic shape and experienced increasing losses toward the end of training. The BERT model was the first to see an increase (epoch 150), followed by RoBERTa (epoch 170) and GPT-2 (epoch 200). This result indicates that the VED models were overfitting the training data - their training loss continued to decrease but validation loss began increasing (Mostafa Ibrahim, 2024). One reason for this is the relatively small size of our training dataset - given the large nature of these models, they could memorize the $20\,000$ training examples fairly quickly (Dolphin, 2023).

These findings helped us extract the best model within each type. For the baseline, we chose the model at the end of training (250 epochs), but for the rest, we chose to stop early, at the lowest point in the validation loss curve. As is evident, the ViT + GPT-2 model performed the best, followed by RoBERTa, BERT and the baseline. Given these results, we proceeded to train a fifth model that replaces our original 224×224 input ViT with one that accepts 384×384 inputs. This encoder was also pretrained on ImageNet-21k, except at a higher resolution. For the decoder, we used GPT-2, which had the lowest validation loss as discussed earlier.

BLEU scores. We tested all five models against our test set, computing BLEU scores (table 1). We see that the ViT + GPT-2 model with 384×384 inputs performed the best, achieving the highest BLEU scores across categories. On average, this model performed 19.3% better than the same configuration using 224×224 inputs. This could be because scientific diagrams tend to use small font in axis labels and legends, which often becomes illegible at a 224×224 resolution. The larger input dimensions allow the model to incorporate this into captions. BERT & RoBERTa models performed similarly, although the latter achieved slightly higher scores. This is likely because the two share similar architectures, although RoBERTa is trained on a larger corpus and uses a dynamic masking strategy (Kumari, 2023). The baseline model performed significantly worse , which is expected given the lack of pre-training and utilization of a ResNet-50 encoder over a ViT (Wu et al., 2020).

	Baseline	**ViT + BERT**	**ViT + RoBERTa**	**ViT + GPT-2**	**ViT + GPT-2** (384×384)
BLEU-1	4.23	14.64	16.45	18.23	21.68
BLEU-2	1.84	10.82	11.64	12.78	16.12
BLEU-3	1.12	7.56	8.12	9.64	11.58
BLEU-4	0.98	4.17	4.89	5.87	6.58

Table 1: BLEU scores on test data for the four models, on a scale of $0 - 100$.

ROUGE scores. Table 2 presents the ROUGE F-1 scores for the models (see section 5.2). We see a similar trend as in BLEU scores, with the ViT + GPT-2 (384×384 input) model performing the best. It should be noted that ROUGE scores are on a scale of 0 to 1, while BLEU scores range from 0 to 100. This is done to match the results presented in prior work. We see that the ROUGE scores are generally below BLEU scores, because our models had worse recall than precision. One reason for this is that the captions generated by the model tended to be more generalized than human ones, often lacking domain-specialized terminology. This is discussed further in section 6.

	Baseline	**ViT + BERT**	**ViT + RoBERTa**	**ViT + GPT-2**	**ViT + GPT-2** (384×384)
ROUGE-1	0.030	0.087	0.086	0.101	0.115
ROUGE-2	0.014	0.057	0.073	0.076	0.098
ROUGE-3	0.012	0.045	0.050	0.065	0.059
ROUGE-4	0.017	0.021	0.042	0.034	0.044

Table 2: ROUGE F-1 scores on test data for the four models on a scale of $0 - 1$.

Comparison with prior work. Given the specialized nature of the task, comparing results with general image captioning would not be an accurate benchmark. The limited research in diagram captioning, such as the work by Hsu et al. (2021), achieved a BLEU-4 score of 2.19 using a ResNet + LSTM model. We see that although are baseline performed slightly worse (likely because we used a small data set), all four VED models beat this result. This is likely due to a combination of pre-training and the advanced capabilities of ViTs. For instance, while ResNet relies on hierarchical feature extraction, ViTs do not by deploying a self attention mechanism (Shabbir, 2023). That said,

our approach still has room for improvement, given that generalized image captioners can achieve BLEU-4 scores of 30 and above (Vinyals et al., 2014). Some pathways are discussed in section 7 .

6 Analysis

To perform a qualitative evaluation, we presented five colleagues with 82 figures and 6 accompanying caption (gold + five model outputs). They rated each caption on a scale of $1 - 10$, with mean scores presented in table 3. The ordering of the average scores is the same as that of BLEU and ROUGE results, although the gold caption received significantly higher ratings than our best model.

	Gold Caption	Baseline	ViT + BERT	ViT + RoBERTa	ViT + GPT-2	ViT + GPT-2 (Large)
Mean	8.78	2.78	4.26	4.48	4.96	5.12

Table 3: Mean scores given by human evaluators, on a scale of 1 to 10.

Besides scores, we also asked our colleagues to briefly comment on each caption, and why they assigned it their respective score. For the baseline, we found that there were significant grammatical errors and a large number of <unk> tokens. This is likely because the transformer decoder was not pre-trained, and hence did not have a good representation of English. Our $N = 20,000$ training samples were insufficient for it to learn coherent grammar, and the limited vocabulary size meant it could not repoduce even mildly technical words like 'sine' and 'parabola'.

For BERT and RoBERTa, the captions were generally coherent with relatively few unknown tokens. However, the comments suggested that the captions were often off topic, describing the general shape of graphs ('a horizontal line' or 'upward bars') rather than the relationship they convey. RoBERTa had fewer instances of this issue (likely because of better masking and a larger pre-training corpus) but tended to have overruns in its captions. For instance, it produced the caption 'error rate against noise power with outgoing radios' for figure 5, where the last four words were completely hallucinated.

The GPT-2 models also suffered from some of the hallucination mentioned above, but generally at a lower frequency. The 384×384 input model did particularly well with complex graphs containing multiple lines, and generally focused on axes and legends when generating captions. A particularly complex input is shown in figure 6, where the model output 'Demodulator bit error rate versus <unk> noise ratio', which is impressive. Nevertheless, testers sometimes found that the model generated a caption that appeared correct but was infact completely irrelevant, which comes as a trade-off with a larger, more vastly pre-trained model. A few examples of captions generated by this model are presented in figure 7.

7 Conclusion

In this paper, we experimented with novel neural methods to address the problem of scientific figure captioning. While previous work in this space utilized CNNs with LSTMs, we presented the first system that incorporates attention-based mechanisms in both the encoder and decoder to beat the state of the art on figure captioning. Our baseline model implemented a ResNet-50 encoder with a vanilla self-attention language model. The low BLEU and ROUGE scores, coupled with poor grammatical structure in most outputs highlight the limitation of approaches that forego pre-training.

In our experiments, we created four Vision Encoder-Decoder models by varying the encoder's input dimensions (224×224 and 384×384) and altering the decoder transformer (BERT, RoBERTa and GPT-2). We found that even with a very small training dataset ($N = 20000$), the pre-training coupled with attention across both the encoder and decoder was sufficient to exceed previous models on both BLEU and ROUGE metrics. The model that worked best was a ViT + GPT-2 architecture accepting 384×384 image inputs, which achived a BLEU-4 score of 6.58 (on a $0 - 100$ scale) and ROUGE-4 F-1 score of 0.044 (on a $0 - 1$ scale). While these scores appear low from a general image captioning perspective, they beat the SOTA in diagram caption by a wide margin.

We also conducted a single-blind human evaluation to uncover the deficiencies of our models. We found that BERT and RoBERTa produced captions that were too simple, while GPT-2 had a tendency to hallucinate irrelevant features. Future work in this space would benefit from using multi-modal encoders that incorporate both the figure's image as well as text (extracted perhaps through OCR). This will likely produce captions that are more relevant and coherent.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering.

Jason Brownlee. 2017. A Gentle Introduction to Calculating the BLEU Score for Text in Python.

Christopher Clark and Santosh Divvala. 2016. Pdffigures 2.0: Mining figures from research papers. In *2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL)*, pages 143–152.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Rian Dolphin. 2023. Overfitting in ML: Avoiding the Pitfalls.

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

Lauren E. Franzblau and Kevin C. Chung. 2012. Graphs, Tables, and Figures in Scientific Publications: The Good, the Bad, and How Not to Be the Latter. *The Journal of Hand Surgery*, 37(3):591–596.

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. A Closer Look at Deep Learning Heuristics: Learning rate restarts, Warmup and Distillation. ArXiv:1810.13243 [cs, stat].

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition.

Ting-Yao Hsu, C. Lee Giles, and Ting-Hao 'Kenneth' Huang. 2021. Scicap: Generating captions for scientific figures.

G Juan. 2021. An Intuitive Explanation of Transformer-Based Models.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Khushboo Kumari. 2023. RoBERTa: A Modified BERT Model for NLP.

Chin-Yew Lin. 2004a. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chin-Yew Lin. 2004b. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows.

Mostafa Ibrahim. 2024. A Deep Dive Into Learning Curves in Machine Learning.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Jeff Wu, R. Child, D. Luan, Dario Amodei, and I. Sutskever. 2019a. Language Models are Unsupervised Multitask Learners.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019b. Language models are unsupervised multitask learners.

Sheeza Shabbir. 2023. 97% Accuracy with ViT on 90 Animal Dataset; A comparative study Vision Transformers vs.

Sun-Joo Shin, Oliver Lemon, and John Mumma. 2001. Diagrams. Last Modified: 2018-12-13.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555.

Eric Wallace. 2020. Speeding Up Transformer Training and Inference By Increasing Model Size.

Yiyu Wang, Jungang Xu, and Yingfei Sun. 2022. End-to-end transformer based model for image captioning.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. 2020. Visual transformers: Token-based image representation and processing for computer vision.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2016. Show, attend and tell: Neural image caption generation with visual attention.

A Appendix

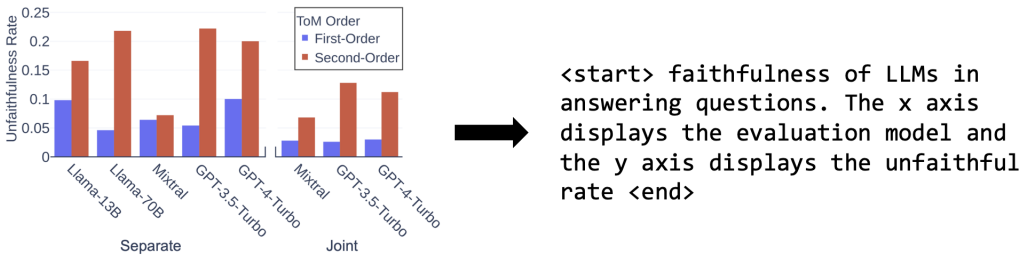


Figure 4: An example input/output pair from our training data.

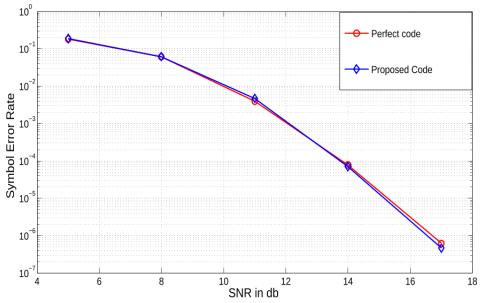


Figure 5: An example input image on which RoBERTa experienced an overrun.

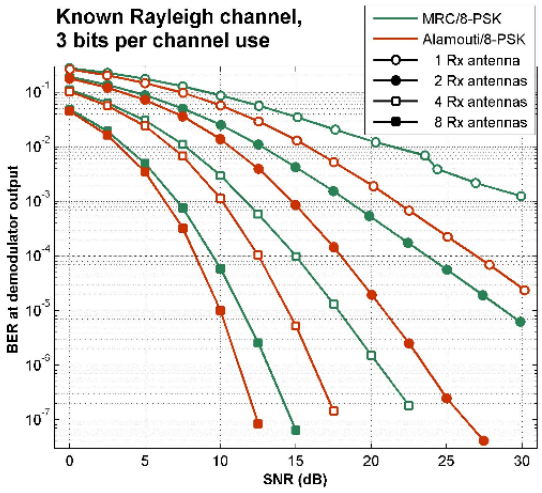
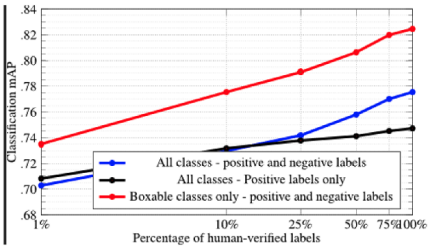
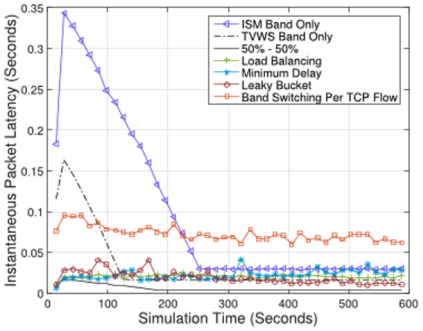


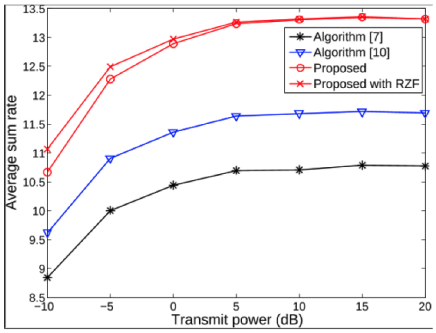
Figure 6: An complex input on which GPT fared well.



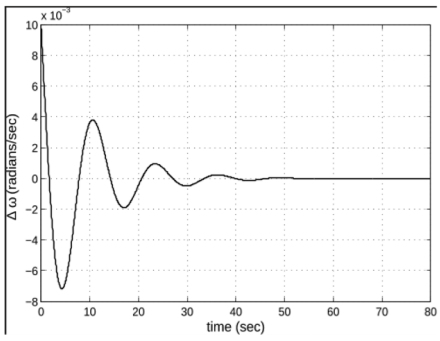
<start> classifier performance versus amount of humanverified labels in terms of percentage of labels <end>



<start> The average number of users for different values of the number of antennas <end>



<start> average sum rate versus transmit power <end>



<start> plot of <unk> for observer versus time to seconds <end>

Figure 7: A few examples of captions generated by ViT + GPT-2 (384×384 inputs)