Fast Parameter Adaptation for Few-shot Image Captioning and Visual Question Answering

Xuanyi Dong[1,2], Linchao Zhu[1,2], De Zhang[3], Yi Yang[1,2]†, Fei Wu[4]
[1]SUSTech-UTS Joint Centre of CIS, Southern University of Science and Technology
[2]CAI, University of Technology Sydney; [3]Information Science Academy, CETC; [4]CCST, Zhejiang University

ABSTRACT

Given only a few image-text pairs, humans can learn to detect semantic concepts and describe the content. For machine learning algorithms, they usually require a lot of data to train a deep neural network to solve the problem. However, it is challenging for the existing systems to generalize well to the few-shot multi-modal scenario, because the learner should understand not only images and texts but also their relationships from only a few examples. In this paper, we tackle two multi-modal problems, i.e., image captioning and visual question answering (VQA), in the few-shot setting.

We propose **F**ast **P**arameter **A**daptation for **I**mage-**T**ext Modeling (FPAIT) that learns to learn jointly understanding image and text data by a few examples. In practice, FPAIT has two benefits. (1) ***Fast learning ability.*** FPAIT learns proper initial parameters for the joint image-text learner from a large number of different tasks. When a new task comes, FPAIT can use a small number of gradient steps to achieve a good performance. (2) ***Robust to few examples.*** In few-shot tasks, the small training data will introduce large biases in Convolutional Neural Networks (CNN) and damage the learner's performance. FPAIT leverages dynamic linear transformations to alleviate the side effects of the small training set. In this way, FPAIT flexibly normalizes the features and thus reduces the biases during training. Quantitatively, FPAIT achieves superior performance on both few-shot image captioning and VQA benchmarks.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**;

KEYWORDS

Few-shot Learning, Image Captioning, Visual Question Answering

ACM Reference Format:
Xuanyi Dong, Linchao Zhu, De Zhang, Yi Yang, Fei Wu. 2018. Fast Parameter Adaptation for Few-shot Image Captioning and Visual Question Answering. In 2018 ACM Multimedia Conference (MM '18), October 22-26, 2018, Seoul, Republic of Korea. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240527

†Corresponding author.

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240527

Figure 1: Image Captioning: given an input image, the model should generate a description of this image. Visual Question Answering: given an open-ended question about an image, the model should generate a natural language answer.

1 INTRODUCTION

It is challenging to build artificial intelligence (AI) systems with human-level intelligence. One of the important abilities of such AI systems is to quickly learn the new concept from only a few examples, especially in the multi-modal scenario. For example, a successful AI system should understand the visual and text inputs as well as their relationship from only a few examples. Recently, researchers have made significant progress in the new concept learning from a few images or videos [8, 10, 29, 34, 37, 40], and words or sentences [22, 41]. However, very few of them study on learning from only a few examples in the multi-modal scenario, e.g., few-shot image captioning and few-shot VQA. The few-shot multi-modal learning problem is more challenging, where both visual and semantic knowledge should be leveraged. Moreover, it is beneficial for real-world applications. For example, Amazon has thousands of new products per week, which contain uncommon words in descriptions and new product images. Jointly understanding images/words and their relationship can better recommend these new products to customers than only focusing on images or focusing on words. In summary, few-shot multi-modal learning is not only challenging but also useful to real-world applications.

In this paper, we tackle few-shot image captioning and VQA to study few-shot multi-modal learning. We show the settings of image captioning and VQA in Figure 1. Image captioning requires the algorithm to generate a description of an image. VQA requires the algorithm to provide an accurate natural language answer given

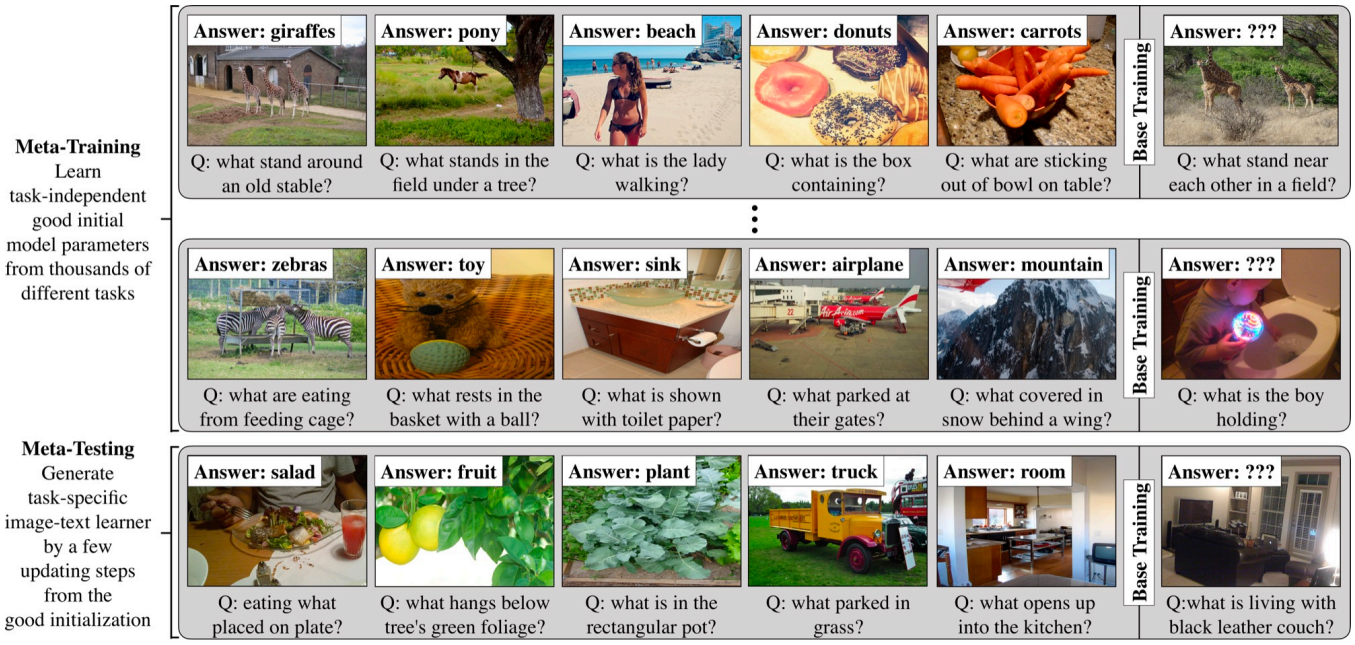


Figure 2: Illustration of FPAIT for visual question answering (VQA). We take the 5-way 1-shot setting for example. Each gray box represents a specific VQA task, which only has five training examples (LEFT) and several testing examples (RIGHT; we only show one testing example for simplification). For this specific task, the answer set in the testing data should be the same with the answer set in the training data. The top row shows the meta-training procedure. During training, FPAIT learns a good initialization of the joint image-text learner from multiple different VQA tasks. This model initialization is task-independent, and it can be quickly adapted into a new different task (base training). The second row shows the meta-testing procedure. Given a new task, FPAIT adapts the initial model to this new task and evaluates on the test set. The final evaluation performance is the average of the testing accuracy on multiple different new tasks. Note that the questions, answers, and images in the tasks of meta-testing are different from that of in meta-training.

a natural language question about an image. For these two tasks, most existing algorithms [30, 36, 44, 46, 50] focus on the supervised setting, and they thus rely on a large amount of human annotated image-text pairs for training. Since some words are uncommon or even unseen in the few-shot scenario, these supervised algorithms cannot well handle such novel words. Some researchers [1, 33, 47] study how to handle novel objects for image captioning and VQA. They usually leverage large external text data, which contains the novel words, to learn the text representation. In this way, their text models can understand the novel words, even if these words are not in training image-text pairs. Compared to this novel object setting, few-shot image captioning and VQA are more difficult, because the target novel (new) objects or words cannot be used in the training data with any form. In this way, their text model [1, 33, 47] cannot learn a good representation for new words, and thus will result in poor performance of image captioning and VQA. Therefore, the existing algorithms might not be eligible for image captioning and VQA tasks in the few-shot setting.

To tackle image captioning and VQA in the few-shot setting, we propose **F**ast **P**arameter **A**daptation for **I**mage-**T**ext Modeling (FPAIT). The main idea of FPAIT is to teach the joint image-text learner ***how to learn***. The classical algorithms [1, 33, 44, 47] usually learn a model to fit the training data. In contrast, FPAIT teaches the learner how to generalize well on a new task using a few training examples quickly. To be specific, FPAIT trains the joint image-text learner on a large number of different visual language tasks with a few examples. As shown in Figure 2, the training objective of FPAIT is to obtain good initial parameters of the joint image-text learner, such that it can maximize the performance of a new task by a few updating steps. Since this initialization is trained from multiple different tasks, the internal representation of this model should be suitable for various tasks, i.e, the model can generalize well. Therefore, FPAIT can achieve good results on new tasks with only a few fine-tuning steps.

Apart from the fast adaptation ability, FPAIT leverages an advanced model to learn the joint image and text representation in the few-shot setting. The existing algorithms [30, 33, 36, 44] usually apply state-of-the-art CNNs to encode image features. These CNN models require a large amount of training data to guarantee a good performance. However, in our settings, there might be only five training examples for one task. Consequently, such small training data will introduce large model bias, which is harmful to the CNN model. To combat the effect of small training data, FPAIT introduces dynamic linear transformations on the image features. These linear transformations are integrated into the CNN model to scale and shift the intermediate features, such that the undesired model

bias can be alleviated. Moreover, the parameters of these linear transformations are generated from the encoded text features, and thereby the text data can effectively influence the image feature. In experiments, the proposed new architecture of FPAIT is superior to the typical image and text embedding methods.

In sum, this paper makes the following contributions:

(1) We propose FPAIT to tackle few-shot image captioning and VQA. FPAIT can train the joint image-text learner, which can quickly adapt itself to a new task given few training examples.

(2) We propose to use an advanced neural network to learn image and text representation in the few-shot scenario jointly. This network generates dynamical parameters from the text data and uses these parameters effectively influence the image encoding.

(3) The proposed FPAIT achieves superior performance on both few-shot image captioning and VQA benchmarks.

2 RELATED WORK

This paper involves the neural image captioning (Sec. 2.1) and the visual question answering (Sec. 2.2). These two problems align with the recent trend of connecting computer vision and natural language [19]. Since we study the few-shot multi-modal learning, this paper is also related to the few-shot learning (Sec. 2.3).

2.1 Image Captioning

Before the success of deep learning [20], image captioning techniques usually utilize hand-crafted features and the performance is limited [9, 21, 28]. Benefit from the rapid development of deep learning [5, 12, 13, 16, 20, 25, 51], researchers have made significant improvements in the image captioning task [26, 43, 44]. The typical approach first encodes the image via CNN into the static visual feature, and then feeds this feature into the recurrent neural network (RNN), e.g., Long Short-Term Memory (LSTM) [14] or Gated Recurrent Units (GRU) [4], to generate natural language descriptions. For example, Xu et al. [44] proposed the attention based model to focus on most attentive regions in images. You et al. [48] proposed to select semantic concept proposals and fuse them into the RNN. Fill-in-the-blank [49] is also considered as an image captioning task, which can be directly solved by the image captioning approaches. The above algorithms achieve impressive image captioning results. However, if there are not enough training image-text pairs, the performance will dramatically decrease. In contrast, FPAIT can quickly adapt the joint image-text learner using a few training examples and result in a good performance.

There are few researchers that study the novel visual concept learning from few language data. Mao et al. [27] proposed a transposed weight sharing scheme to prevent the overfitting problem in the new concept learning. They only evaluated on three novel concepts, and the test dataset is quite small compared to the benchmark image captioning dataset, i.e., MSCOCO [24]. Therefore, it is unclear about their effect on the large-scale dataset, especially when evaluating the algorithm on a substantial number of novel concepts. Yao et al. [47] incorporated the copying mechanism into LSTM to describe novel objects in captions. However, these methods [1, 47] use unpaired text training data to learn the representation of novel objects. In this way, their text models will learn the concept of the novel objects, which can not be considered as few-shot. Therefore, their algorithm cannot deal with new objects that are unseen in both image and text data.

2.2 Visual Question Answering

VQA has attracted an increasing interest since Antol et al. [2]. A simple baseline algorithm for VQA consists of two components that (1) feed the concatenation of both the question text feature and the image visual feature into RNN; (2) decode the concatenated feature into the output answers [2]. It is essential for VQA models to leverage how to combine image feature and text feature. A straightforward extension of the simple concatenation is joint embedding. Many approaches follow this joint embedding idea [30, 36, 46, 52]. For example, Noh et al. [30] proposed to learn a CNN with a dynamic parameter layer for VQA. Teney et al. [39] proposed to improve VQA with structured representations of both scene contents and questions. Hu et al. [15] proposed end-to-end module networks to predict answers by reasoning. These supervised methods require many manually labeled image-question-answer pairs for training. Therefore, these approaches are not suitable to the cases, in which only few training examples are available.

To tackle the novel object learning problem in VQA, some algorithms [33] utilize the external text, e.g., book or Wikipedia, to learn the word representation of novel objects. However, we study a more difficult setting that the novel concept cannot appear in any pre-training data. Teney et al. [38] leveraged a similarity-based meta-learning approach for few-shot VQA. They only evaluate the algorithm on the small VQA-Number dataset [11], in which the number of classes is only seven. Therefore, they lack the evidence of the effectiveness on the large-scale datasets.

2.3 Few-shot learning

The existing optimization algorithms require many labeled data to update deep CNNs. Therefore, they perform poorly when only few training examples are available. However, the model excelling at learning from few examples can be useful in practical applications, and this motivates the research on few-examples learning [6, 7, 42] and few-shot learning [10, 29, 45]. Few-example learning allows the use of unlabeled data, while few-shot learning not. In this paper, we focus on the few-shot learning setting.

There are several types of notable methods for few-shot learning. Finn et al. [10] constrained the meta-learner to use ordinary gradient descent to update the base-learner. Munkhdalai et al. [29] investigated sophisticated weight update scheme for the few-shot classification model. The most related work to ours is Finn et al. [10], which only takes images as inputs. In contrast, ours can handle the integrated image and text pair input. Moreover, to combat the side effects of small training examples in the multi-modal scenario, we use the text feature to generate dynamic parameters to normalize the image features automatically.

3 METHODOLOGY

FPAIT is motivated by the recent techniques in meta-learning [10, 37] and conditional normalization [17, 31]. Meta-learning usually focuses on few-shot image classification, whereas FPAIT extends it into image and text joint modeling. Moreover, FPAIT takes the

advantages of conditional normalization to overcome the model bias, which is introduced by few training examples.

3.1 Preliminary

For image captioning, we use the fill-in-the-blank setting [49]. This kind of image captioning asks the algorithm to fill in the blank of a description template for an image. For example, one description template can be "The frisbee is [blank]", which requires to fill in the "[blank]" with the appearance of frisbee in a corresponding image. This problem can be formulated as: given a fill-in-the-blank description template $\mathbf{Q}$ and an image $\mathbf{I}$, we need a function f that takes $\mathbf{Q}$ and $\mathbf{I}$ as inputs to generate words or phrases $\mathbf{A}$ for this blank. For VQA, given a natural language question about an image, we need a function to generate the natural language answer to the question. For the simplicity of notation, we denote the question as $\mathbf{Q}$, the function as $\mathbf{f}$, and the image as $\mathbf{I}$ for VQA. Suppose that there are C different classes of answers, the function $\mathbf{f}$ can be a neural network that maps $\mathbf{Q}$ and $\mathbf{I}$ to the confidence scores over these C candidate answers [46, 49]. In this way, the fill-in-the-blank and VQA models can be formulated as: $\mathbf{p} = \mathbf{f}(\mathbf{Q}, \mathbf{I})$, where $\mathbf{p} \in \mathcal{R}^C$ is the confidence score vector over C classes. Most existing algorithms usually take $\mathbf{y} = arg\ \max_i \mathbf{p}_i$ as the the final prediction, where $\mathbf{p}_i$ denotes the i-th element of $\mathbf{p}$ and $1 \le i \le C$.

Traditional image captioning and VQA settings. Suppose the training set is $\mathcal{T}^{train} = \{(\mathbf{Q}_i, \mathbf{I}_i, \mathbf{A}_i) \mid 1 \le i \le n\}$ and the testing set is $\mathcal{T}^{test} = \{(\mathbf{Q}_i, \mathbf{I}_i, \mathbf{A}_i) \mid 1 \le i \le m\}$. Most image captioning or VQA methods learn the parameters θ of $\mathbf{f}$ by maximizing the likelihood of the correct answers or blank words:

$$\theta = arg\ \max_{\theta} \sum_{i=1}^{n} \log P(\mathbf{y}_i = \mathbf{A}_i | \mathbf{f}_\theta(\mathbf{Q}_i, \mathbf{I}_i)), \tag{1}$$

where i is the index of the training data, and P denotes the conditional probability. To evaluate the performance, $\mathbf{f}_\theta$ is applied to each testing data to obtain the predicted answers or descriptions. The evaluation performance is calculated by some evaluation metrics [24], e.g., METEOR for image captioning and WUPS for VQA.

3.2 Fast Parameter Adaptation

Instead of the traditional setup for Eq. (1), we consider the meta-learning setup [10]. It has a distribution over tasks $p(\mathcal{T})$[1] and requires the learned model can achieve good performance on different task samples from $p(\mathcal{T})$, rather than a single individual task in the traditional setup. In the K-shot learning setting, a new task $\mathcal{T}$ sampled from $p(\mathcal{T})$ has only K training samples. Take the N-way classification problem as an example, each task $\mathcal{T}$ will contain $K \times N$ training examples, i.e., K examples for each class. To solve this meta-learning problem, FPAIT learns a good initialization of $\mathbf{f}_\theta$ to be able to quickly adapt to different tasks, which are sampled from $p(\mathcal{T})$. The trained joint image-text learner $\mathbf{f}_\theta$ should quickly adapt its parameters based on these K training samples, and then generalize well on new samples from $\mathcal{T}$. After the parameters of $\mathbf{f}_\theta$ are adapted into a specific $\mathcal{T}$, we will evaluate $\mathbf{f}_\theta$ on new examples sampled from this $\mathcal{T}$.

[1]Suppose the number of answer classes in the training set of the traditional setup is C, a task sampled from $p(\mathcal{T})$ in the N-way K-shot meta-learning setup is that (1) randomly sample N classes from the total C classes; (2) for these sampled N classes, randomly sample $2 \times K$ data-points for each class; (3) the training set of this task consists of the first K data-points and the testing set consists of the last K data-points.

Algorithm 1 Meta-Training Procedure of FPAIT (N-way K-shot)

Input: $p(\mathcal{T})$: distribution over tasks
Input: *net*: the joint image-text learner with the parameter θ
1: initialize θ
2: **while** not converge **do**
3: Sample batch of tasks, in which each task is $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Sample $K \times N$ examples $\mathcal{D}_i = \{\mathbf{I}^{(j)}, \mathbf{Q}^{(j)}, \mathbf{A}^{(j)}\}$ from $\mathcal{T}_i$
6: Compute gradient and adapt θ into $\theta_i^{'}$ via Eq. (2).
7: Sample $K \times N$ new examples $\mathcal{D}_i^{'} = \{\mathbf{I}^{(j)}, \mathbf{Q}^{(j)}, \mathbf{A}^{(j)}\}$ from $\mathcal{T}_i$
8: **end for**
9: Compute the meta gradient $\nabla_\theta \mathcal{L}_{\mathcal{D}_i^{'}}(\mathbf{f}_{\theta_i^{'}})$ for θ
10: Update θ based on the meta gradient via Adam to optimize Eq. (3)
11: **end while**
Output: a good initialization parameter θ

Formally, FPAIT aims to learn good initial parameters θ of the joint image-text learner $\mathbf{f}_\theta$, such that $\mathbf{f}_\theta$ can be fine-tuned by a small number of gradient-based updating steps on the new tasks. To be formal, given a new task $\mathcal{T}_i$, we denote the parameters as θ, and the parameters after adaptation as $\theta^{'}$. Suppose that we use one gradient update of the basic Stochastic Gradient Descent (SGD) [32] for the parameter adaptation, then $\theta^{'}$ is computed as:

$$\theta_i^{'} = \mathbf{Opt}(\theta, \mathcal{T}_i) = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\mathbf{f}_\theta), \tag{2}$$

where α denotes the learning step size, $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\mathbf{f}_\theta)$ represents the gradient of parameters θ *w.r.t.* the loss function $\mathcal{L}$, and $\mathcal{L}$ is the likelihood loss in Eq. (1). Here, we use one gradient step for the notation simplicity, whereas the parameter adaptation algorithm **Opt** can also use multiple gradient steps of SGD or Adam [18].

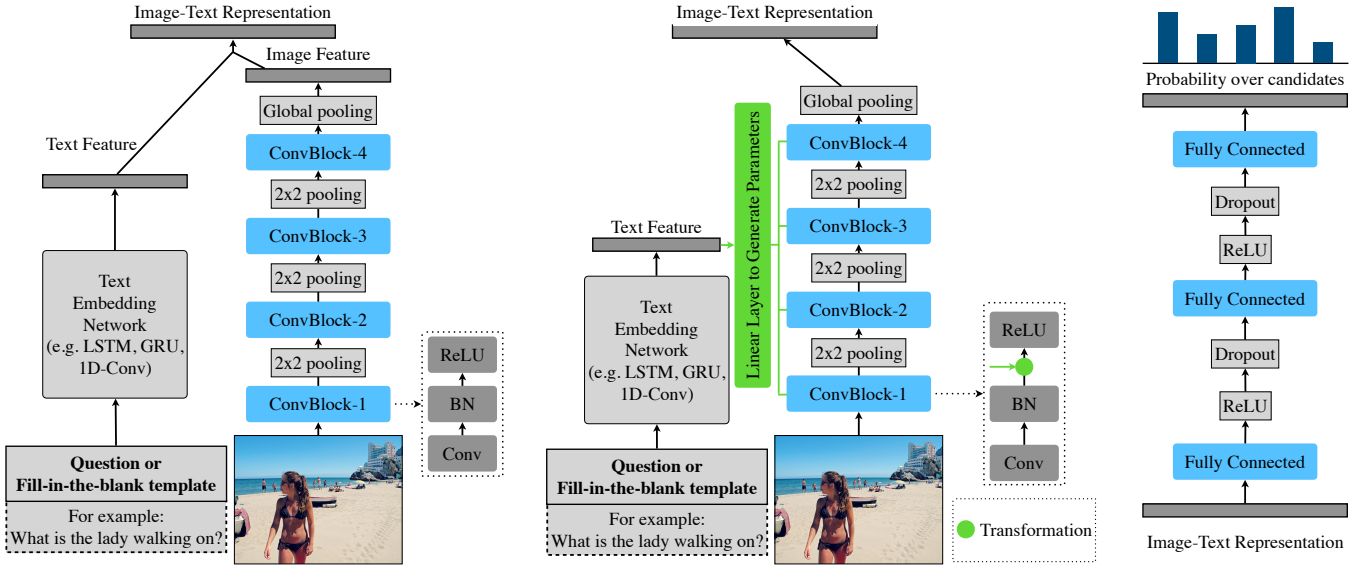
We show the meta-training procedure in Figure 2. During training, we optimize the test error of the adapted parameters $\theta_i^{'}$ for various of different tasks $\mathcal{T}_i$ from the distribution $p(\mathcal{T})$. Such objective function can be formulated as :

$$\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(\mathbf{f}_{\theta_i^{'}}) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(\mathbf{f}_{\mathbf{Opt}(\theta, \mathcal{T}_i)}), \tag{3}$$

This objective is computed based on the adapted parameters θ_i, but we should notice that it will finally optimize the parameters θ of the joint image-text learner. Therefore, FPAIT can optimize θ to generalize well on a new task after several gradient-based update steps. The gradient of the parameters θ in Eq. (3) is computed as $\nabla_\theta \sum_{\mathcal{T}_i\ p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(\mathbf{f}_{\theta_i^{'}})$. Using this gradient, we can apply any gradient-based optimization algorithms to update the parameters θ, such as Nesterov SGD or Adam.

3.3 FPAIT Architecture and Algorithm

Encode Image. To leverage the advantage of deep learning, we use CNN to encode images into features [13, 16]. Similar to [10, 37], we design a small CNN to generate the image features. As shown in Figure 3, it consists of four convolutional blocks, in which each



(a) Baseline model for the joint image-text modelling **(b) FPAIT for the joint image-text modelling** **(c) Prediction model**

Figure 3: The joint image-text learner in FPAIT consists of two parts, the representation module and the prediction module. The representation module jointly encodes both image and text inputs into one feature vector. The typical way is to concatenate the image CNN feature and text embedding feature, as Figure 3(a). In contrast, FPAIT uses text features to dynamically generate parameters for a transformation function. This transformation function can adjust the intermediate features in the CNN model. In this way, FPAIT can effectively model both image and text information, and combat the side effects of a few training examples in few-shot learning. The prediction module takes the image-text representation as input to generate the final classification probability.

block has a 3×3 convolutional layer with batch normalization (BN) and ReLU. The feature channels in these four blocks are 64, 96, 128, 256, respectively. There are three pooling layer with stride of 2 after the first three convolutional blocks, and one global pooling layer after the last convolutional block. Therefore, the output of this CNN is a 256-dimension feature vector.

Encode Text. In the image captioning task (fill-in-the-blank), the text input is the fill-in-the-blank description template. For example, a template can be "<blank> is walking on the beach" for the image in Figure 3. In the VQA task, the text input is a natural language question. The typical way to encode text is first to use the word embedding to transfer each word into a feature vector, and then input these features into an RNN sequentially. Some researchers also use temporal convolutional network [23] to replace RNN. In this paper, we compare different kinds of encoders for text in the experiment section, and we constrain the output of these encoders to be a 512-dimension feature vector.

FPAIT. The joint image-text learner needs to fuse the text features and image features effectively. A classical fusion method is to concatenate the image feature and text feature, as shown in Figure 3(a). However, this kind of model requires a large amount data to train, but few-shot learning only has a few training examples. It will affect the performance due to the large model bias introduced by the small number of training examples. Moreover, in the multi-modal scenario, the model should not only learn the image/text information but also their relationship, in which the model bias problem becomes more severe. The classical fusion method, e.g., concatenation, product, and attention, might not be robust to such few-shot multi-modal scenario. To alleviate the model bias problem in few-shot multi-modal scenario, we apply a transformation function with dynamic parameters. We have multiple choices for this transformation function, such as a convolutional layer or a fully connected layer. In this paper, we choose a Channel-wise Linear Transformation (CLT). Suppose the text encoder as $\mathbf{g}$, it takes the text $\mathbf{Q}$ as input to generate the text feature $\mathbf{g}(\mathbf{Q})$. For the features of the c-th layer in CNN, the parameters of CLT are γ_c and β_c as follows:

$$\gamma_c = \mathbf{g}_\gamma(\mathbf{g}(\mathbf{Q})) \; ; \; \beta_c = \mathbf{g}_\beta(\mathbf{g}(\mathbf{Q})), \tag{4}$$

$$CLT(\mathbf{F}_c) = \mathbf{F}_c \times \gamma_c + \beta_c, \tag{5}$$

where $\mathbf{g}_\gamma$ and $\mathbf{g}_\beta$ are fully connected layers. $\mathbf{F}_c$ denotes the features of the c-th CNN layer. γ_c and β_c are two vectors, of which the length is the same with the channel of $\mathbf{F}_c$. As shown in Figure 3(b), we apply the CLT to the features after each BN layer. In this way, the image-text representation is the CNN output.

To predict the final output of FPAIT, we use a prediction part with three fully connected layers, as shown in Figure 3(c). The output dimensions of the first two fully connected layers are 512 and 512. The last fully connected layer outputs the probability over the candidate answers or blanks.

Overall Algorithm. The image-text learner in FPAIT consists of two parts, i.e., Figure 3(b) and Figure 3(c). We denote this model as $\mathbf{f}$ that takes the image $\mathbf{I}$ and text $\mathbf{Q}$ as inputs. The loss function of base task is a cross-entropy loss between the prediction $\mathbf{f}(\mathbf{I}, \mathbf{Q})$

and the ground truth answer/blank **A**. The training algorithm is shown in Algorithm 1.

Discussions about Pre-training. We do not use the ImageNet [20] to pre-train the CNN. The image classes of ImageNet contain the visual concepts in image captioning or VQA datasets. If we use ImageNet for pre-training, the CNN will know many concepts, e.g., dolphin and dog. It is possible that these concepts are the target new concepts in image captioning or VQA tasks. In this case, it can not be considered as few-shot setting.

Discussions about CNN model in FPAIT. We follow [10, 37] to use a similar small CNN model rather than large CNNs, such as ResNet-101 [13]. The large CNN models can easily overfit the few-shot training examples and be harmful to the performance.

4 EXPERIMENTAL EVALUATION

We evaluate the proposed algorithm on two benchmark datasets, i.e., Toronto COCO-QA [35] for VQA and MSCOCO Captioning [24] for image captioning. We first introduce the details of these datasets in Section 4.1 and the experimental settings in Section 4.2. Later, we compare with some state-of-the-art algorithms [44] in Section 4.3. Lastly, qualitative results and analysis are shown in Section 4.4.

4.1 Benchmark Datasets

Toronto COCO-QA [35] contains 78,736 training questions, 38,948 testing questions, and 123,287 images in total. Each question is associated with one answer and one image. Each question is also labeled with one QA type, and there are four different QA types, i.e., object, number, color, and location. We use the following five steps for pre-processing to clean up the questions and answers: ***(1)*** Merge the official training and testing sets into one VQA set, denoted as $\mathcal{T}^{VQA-1}$. ***(2)*** From $\mathcal{T}^{VQA-1}$, select the image-question pairs, which is labeled by the object type, as $\mathcal{T}^{VQA-2}$. ***(3)*** From $\mathcal{T}^{VQA-2}$, select the image-question pairs, in which the answer occurs more than four times, as $\mathcal{T}^{VQA-3}$. ***(4)*** From $\mathcal{T}^{VQA-3}$, select every image-question pair, in which all words in this question occur more than four times in all words of $\mathcal{T}^{VQA-3}$, as $\mathcal{T}^{VQA-4}$. ***(5)*** $\mathcal{T}^{VQA-4}$ is a clean VQA dataset. We randomly select eighty percent of this clean VQA dataset as the training set and the rest of this clean VQA dataset as the testing set. Consequently, there are 57,834 questions in the training set with 256 different kinds of words in the answer; there are 13,965 questions in the testing set with 65 different kinds of words in the answer. **Note that** if one word is in the answers of the training set, then any answer of the testing set cannot contain this word.

MSCOCO Captioning [24] contains 82,783 training, 40,504 validation, and 40,775 testing images. Each image has around 5 crowd-sourced captions. We pre-process MSCOCO Captioning to generate the fill-in-the-blank dataset, denoted as COCO-FITB. There are four steps. ***(1)*** Prepare image-caption pairs. We use the processed captioning data from Lu et al. [26], which has 616,767 image-caption pairs. ***(2)*** Collect candidate blank words **B**. Following Lu et al. [26], we use their manually selected 413 fine-grained classes for the candidate "blank"s. We select classes with only one word as the candidate blank words **B**. ***(3)*** Generate image and description template pairs. We first select the image-caption pairs with only one word in the candidate blanks **B**. Then, if a word in captions is in **B**, this word will be replaced by "<blank>". ***(4)*** We randomly select eighty percent of the filtered dataset as the training set, and the rest as the testing set. Lastly, there are 181,844 image-caption pairs with 159 blank word classes for training, 34,919 image-caption pairs with 43 blank word classes for testing. **Note that** the blank word indicates the ground truth labeled to fill in the blank. The blank words in the training set are different from that of in the testing set. Besides, blank words are not in the captions.

Table 1: Statistics on two experimental benchmarks.

		Toronto COCO-QA	COCO-FITB
Task		VQA	Image Captioning
#Pair	Meta-Train	57,834	181,844
	Meta-Testing	13,965	34,919
#Class	Meta-Train	256	159
	Meta-Testing	65	43

Table 1 shows the statistics on two benchmarks, which are public available at GitHub[2]. These two datasets are much larger than previous few-shot VQA datasets.

4.2 Experimental Settings

Few-shot VQA and image captioning setup. In few-shot image captioning (fill-in-the-blank), given a task $\mathcal{T}$ with a few training examples, the joint image-text learner should predict the blank for a description template. In few-shot VQA, given a task $\mathcal{T}$ with a few training examples, the joint image-text learner should predict an answer for an image-question pair. Following the common setting in few-shot learning [10, 37, 40], we use the N-way K-shot setting, where $N \in \{5, 10, 20\}$ and $K \in \{1, 5\}$. For N-way K-shot fill-in-the-blank, there are N different kinds of blanks, and each blank class has K training examples. For N-way K-shot VQA, there are N different kinds of answers, where each answer class has K training examples. Therefore, we view both image captioning and VQA as the classification problem. Given an image-question pair for VQA or an image-template pair for fill-in-the-blank, the joint image-text learner takes the image and text as inputs to predict N confidence scores, i.e., the probability of being the i-th blank/answer class.

Details of Compared Algorithms. We compare our algorithm with **six** different state-of-the-art methods in Table 2 and Table 3. These compared algorithms can be categorized into two different classes. (1) One is to first pre-train the model on the training setting in the classical supervised learning, denoted as pre-training task. Given one new task, the last classification layer will be randomly initialized since the classes are different between the pre-training task and the new task. Then we fine-tune the model on the new task with only few training examples and evaluate its performance. This kind of method is indicated as "Fine-tuning" in Table 2 and Table 3. (2) The other is using the Algorithm 1 to train the model in the meta-learning setting, as introduced in Section 3.2. We indicate the training set in the meta-learning setting as the meta-training set.

Architectures. For different methods, we use the same CNN architecture as introduced in Section 3.3 to make a fair comparison.

[2]https://github.com/D-X-Y/FPAIT

Table 2: Comparison of accuracy on Toronto COCO-QA for few-shot visual question answering. "w/o" indicates without.

Toronto COCO-QA [35]	5-way accuracy		10-way accuracy		20-way accuracy	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
CNN+RNN (Fine-tuning)	52.11	65.03	40.38	51.52	28.06	37.17
CNN+GRU (Fine-tuning)	53.38	66.12	42.71	53.99	29.24	40.56
CNN+LSTM (Fine-tuning)	53.50	66.65	40.33	54.55	29.99	43.05
CNN+TCN (Fine-tuning)	57.19	71.82	44.96	58.83	33.66	47.17
CNN+TCN+CLT (Fine-tuning)	56.72	70.10	43.76	59.19	32.92	46.97
FPAIT w/o CLT	59.38	71.92	45.11	60.20	34.09	47.91
FPAIT	**60.61**	**72.17**	**46.37**	**60.92**	**34.54**	**48.20**

Table 3: Comparison of accuracy on COCO-FITB for few-shot image captioning. "w/o" indicates without.

COCO-FITB [24]	5-way accuracy		10-way accuracy		20-way accuracy	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
CNN+RNN (Fine-tuning)	51.14	61.37	37.92	49.18	25.97	34.74
CNN+GRU (Fine-tuning)	53.16	62.34	39.16	50.92	29.64	37.29
CNN+LSTM (Fine-tuning)	50.12	62.39	39.30	49.50	28.53	37.10
CNN+TCN (Fine-tuning)	59.95	70.32	46.59	58.41	36.26	45.67
CNN+TCN+CLT (Fine-tuning)	57.89	69.84	45.39	57.94	34.89	44.42
FPAIT w/o CLT	60.13	70.88	47.10	59.31	37.09	46.91
FPAIT	**61.01**	**71.13**	**47.79**	**60.91**	**38.17**	**47.32**

To encode the text information, we analyze five different kinds of models as follows: (1) The basic one-layer RNN with hidden state dimension of 512 and dropout ratio of 0.5, denoted as RNN. (2) The one-layer LSTM [14] with hidden state dimension of 512 and dropout ratio of 0.5, denoted as LSTM. (3) The one-layer GRU [4] with hidden state dimension of 512 and dropout ratio of 0.5, denoted as GRU. (4) The one-block TCN [3] with hidden state dimension of 512 and dropout ratio of 0.5, denoted as TCN. (5) The TCN architecture integrated with the CLT module in FPAIT, denoted as TCN+CLT. Note that in Table 2 and Table 3, CNN+TCN+CLT has the same architecture with FPAIT, and CNN+TCN has the same architecture with "FPAIT w/o CLT". However, CNN+TCN+CLT and CNN+TCN use the fine-tuning training strategy, but FPAIT and "FPAIT w/o CLT" use the meta-learning training strategy.

4.3 Results on VQA and Image Captioning

Training Strategy. To train the fine-tuning method, we first train the network on the training set by Adam with 50 epochs. The learning rate is 0.001 and the batch size is 32. When the new task with $N \times K$ training examples comes, we only retrain the last classification layer of the network and fix the rest of the network. When retraining, we use the Adam optimizer with learning rate of 0.001 and 100 epochs. To train FPAIT and "FPAIT w/o CLT", we use the baseline model as the initialization (step 1 in Alg. 1), and then we use the SGD with learning rate of 0.01 and step of 5 for the inner loop updating (Eq. (2)). For the meta-training, we use Adam with batch size of 32 and learning rate of 0.001 to optimize the model (step 10 in Algorithm 1). For Toronto COCO-QA and COCO-FITB, we use the same training strategies.

Results on Image Captioning. We show the comparison results on MSCOCO Captioning in Table 3. For the fine-tuning methods, CNN+RNN is the simplest model and achieves the worst performance. The performance of CNN+GRU is similar to the performance of CNN+LSTM, and is higher than CNN+RNN by about 1% absolute accuracy. The models, which have temporal convolutional neural network to encode the text representation, are CNN+TCN and CNN+TCN+CLT. These two models achieve much higher performance compared to CNN with RNN/GRU/LSTM. On average, the accuracies of CNN+TCN are higher than CNN+LSTM by about 5% absolute accuracy. CNN+TCN obtains a slightly higher accuracy than CNN+TCN+CLT. This can be caused by that, under the fine-tuning setting, CNN+TCN+CLT overfits the pre-training set, and thus the feature learned by CNN+TCN+CLT has a worse generalization ability. In Table 3, FPAIT is superior to all compared algorithms. This implies that FPAIT is more suitable for the few-shot multi-modal scenario than fine-tuning.

Results on VQA. We show the comparison of accuracies on COCO-FITB in Table 2. We can obtain the same conclusion as in Toronto COCO-QA. (1) The meta-learning algorithm used in FPAIT is superior to the fine-tuning method. (2) The temporal convolution neural network is superior to RNN for modeling the text representation in the few-shot multi-modal scenario. (3) FPAIT achieves the best performance compared to all compared algorithms.

Fast learning ability. FPAIT can quickly adapt the learned good initialization into a new task by only a few gradient steps (usually five steps in experiments). Compared to the baseline methods, which requires about 100 gradient steps for adaptation, FPAIT is more efficient and can learn fast.

Robust to few examples. In experiments, FPAIT can use one training sample per class to train a robust model. By evaluating on thousands of different samples, FPAIT can achieve the accuracy

Figure 4: Qualitative results of FPAIT on the Toronto COCO-QA testing set.

of about 50%. This validates that FPAIT is robust to few training examples.

4.4 Qualitative Analysis

We show the qualitative results of FPAIT in Figure 4. We should notice that few-shot multi-modal learning is very difficult. During training, in the fifth examples of the i-th task, the carrot (the answer of the question) is blocked by the hand and not easy to recognize, but the model must quickly learn how to reason from the question to a tiny region in the image. In the first testing example, the question is "what is the hand feeding to a giraffe", and the model needs to understand the question and then find the corresponding region in the image as well as recognizing the object. Note that the model can only see one carrot example during training, but should learn both carrot representation and its relationship to questions. From the visualized results, FPAIT can associate the image and text inputs, and predict an accurate answer well. One failure case is about "helicopter". The training helicopter example is a black large one in the warehouse. However, the testing helicopter example is a tiny red one with a different model flying in the sky. FPAIT fails because this example is quite different from the training examples.

5 CONCLUSION

We propose FPAIT for the few-shot multi-modal learning. FPAIT leverages a fast parameter adaptation algorithm to train an joint image-text learner in the few-shot multi-modal scenario. In this way, FPAIT can learn a good parameter initialization, such that the learner can quickly adapt to new tasks using a few gradient-based updating steps. The classical models for image and text require a large amount of training data. When the training set is extremely small, the performance of these models will significantly degenerate. To alleviate the side effects of the small training set, FPAIT equips the visual model with dynamic linear transformations, of which parameters are generated from text features. In experiments, we focus on few-shot image captioning and few-shot VQA. On both tasks, FPAIT is superior to the state-of-the-art algorithms.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China (Nos. 61625107, U1611461, U1509206), the Key Program of Zhejiang Province, China (No. 2015C01027), and partially supported by an Australian Research Council Discovery Project. We acknowledge the Data to Decisions CRC (D2D CRC) and the Cooperative Research Centres Programme for funding this research.

REFERENCES

[1] Lisa Anne Hendricks, Subhashini Venugopalan, Marcus Rohrbach, Raymond Mooney, Kate Saenko, Trevor Darrell, Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, et al. 2016. Deep Compositional Captioning: Describing Novel Object Categories without Paired Training Data. In *CVPR*.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *ICCV*.

[3] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv preprint arXiv:1803.01271* (2018).

[4] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *EMNLP*.

[5] Xuanyi Dong, Junshi Huang, Yi Yang, and Shuicheng Yan. 2017. More Is Less: A More Complicated Network With Less Inference Complexity. In *CVPR*.

[6] Xuanyi Dong, Deyu Meng, Fan Ma, and Yi Yang. 2017. A dual-network progressive approach to weakly supervised object detection. In *ACM on Multimedia*.

[7] Xuanyi Dong, Shoou-I Yu, Xinshuo Weng, Shih-En Wei, Yi Yang, and Yaser Sheikh. 2018. Supervision-by-Registration: An Unsupervised Approach to Improve the Precision of Facial Landmark Detectors. In *CVPR*.

[8] Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. 2018. Few-Example Object Detection with Model Communication. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018). https://doi.org/10.1109/TPAMI.2018.2844853

[9] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. 2010. Every Picture Tells a Story: Generating Sentences from Images. In *ECCV*.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*.

[11] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-level Performance on ImageNet Classification. In *ICCV*.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* (1997).

[15] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to Reason: End-To-End Module Networks for Visual Question Answering. In *ICCV*.

[16] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. 2017. Densely Connected Convolutional Networks. In *CVPR*.

[17] Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-Time With Adaptive Instance Normalization. In *ICCV*.

[18] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual Genome: Connecting Language and Vision using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision* (2017).

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*.

[21] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. 2013. BabyTalk: Understanding and Generating Simple Image Descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2013). https://doi.org/10.1109/TPAMI.2012.162

[22] Angeliki Lazaridou, Marco Marelli, and Marco Baroni. 2017. Multimodal Word Meaning Induction from Minimal Exposure to Natural Text. *Cognitive Science* (2017).

[23] Colin Lea, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. 2017. Temporal Convolutional Networks for Action Segmentation and Detection. In *CVPR*.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in Context. In *ECCV*.

[25] Yu Liu, Fangyin Wei, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. 2018. Exploring Disentangled Feature Representation beyond Face Identification. In *CVPR*.

[26] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2018. Neural Baby Talk. In *CVPR*.

[27] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2015. Learning Like a Child: Fast Novel Visual Concept Learning from Sentence Descriptions of Images. In *ICCV*.

[28] Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daumé III. 2012. Midge: Generating Image Descriptions from Computer Vision Detections. In *EACL*.

[29] Tsendsuren Munkhdalai and Hong Yu. 2017. Meta Networks. In *ICML*.

[30] Hyeonwoo Noh, Paul Hongsuck Seo, and Bohyung Han. 2016. Image Question Answering using Convolutional Neural Network with Dynamic Parameter Prediction. In *CVPR*.

[31] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. FiLM: Visual Reasoning with a General Conditioning Layer. In *AAAI*.

[32] Ning Qian. 1999. On the momentum term in gradient descent learning algorithms. *Neural Networks* (1999).

[33] Santhosh K Ramakrishnan, Ambar Pal, Gaurav Sharma, and Anurag Mittal. 2017. An Empirical Evaluation of Visual Question Answering for Novel Objects. In *CVPR*.

[34] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a Model for Few-shot Learning. In *ICLR*.

[35] Mengye Ren, Ryan Kiros, and Richard Zemel. 2015. Exploring Models and Data for Image Question Answering. In *NIPS*.

[36] Kevin J Shih, Saurabh Singh, and Derek Hoiem. 2016. Where to Look: Focus Regions for Visual Question Answering. In *CVPR*.

[37] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical Networks for Few-shot Learning. In *NIPS*.

[38] Damien Teney and Anton van den Hengel. 2017. Visual Question Answering as a Meta Learning Task. *arXiv preprint arXiv:1711.08105* (2017).

[39] Damien Teney, Lingqiao Liu, and Anton van den Hengel. 2017. Graph-Structured Representations for Visual Question Answering. In *CVPR*.

[40] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. 2016. Matching Networks for One Shot Learning. In *NIPS*.

[41] Su Wang, Stephen Roller, and Katrin Erk. 2017. Distributional Modeling on a Diet: One-shot Word Learning from Text only. In *IJCNLP*.

[42] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. 2018. Exploit the Unknown Gradually: One-Shot Video-Based Person Re-Identification by Stepwise Learning. In *CVPR*.

[43] Yu Wu, Linchao Zhu, Lu Jiang, and Yi Yang. 2018. Decoupled Novel Object Captioner. In *ACM on Multimedia*.

[44] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*.

[45] Zhongwen Xu, Linchao Zhu, and Yi Yang. 2017. Few-shot object recognition from machine-labeled web images. In *CVPR*.

[46] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked Attention Networks for Image Question Answering. In *CVPR*.

[47] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2017. Incorporating Copying Mechanism in Image Captioning for Learning Novel Objects. In *CVPR*.

[48] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image Captioning with Semantic Attention. In *CVPR*.

[49] Licheng Yu, Eunbyung Park, Alexander C Berg, and Tamara L Berg. 2015. Visual Madlibs: Fill in the Blank Description Generation and Question Answering. In *ICCV*.

[50] Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. 2018. Learning to Count Objects in Natural Images for Visual Question Answering. In *ICLR*.

[51] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. 2018. Camera Style Adaptation for Person Re-Identification. In *CVPR*.

[52] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *CVPR*.