




Article

VL-Meta: Vision-Language Models for Multimodal Meta-Learning

Han Ma , Baoyu Fan , Benjamin K. Ng * and Chan-Tong Lam 

Faculty of Applied Sciences, Macao Polytechnic University, Macao 999078, China;
han.ma@mpu.edu.mo (H.M.); baoyu.fan@mpu.edu.mo (B.F.); ctlam@mpu.edu.mo (C.-T.L.)
* Correspondence: bng@mpu.edu.mo

Abstract: Multimodal learning is a promising area in artificial intelligence (AI) that can make the model understand different kinds of data. Existing works are trying to re-train a new model based on pre-trained models that requires much data, computation power, and time. However, it is difficult to achieve in low-resource or small-sample situations. Therefore, we propose VL-Meta, Vision Language Models for Multimodal Meta Learning. It (1) presents the vision-language mapper and multimodal fusion mapper, which are light model structures, to use the existing pre-trained models to make models understand images to language feature space and save training data, computation power, and time; (2) constructs the meta-task pool that can only use a small amount of data to construct enough training data and improve the generalization of the model to learn the data knowledge and task knowledge; (3) proposes the token-level training that can align inputs with the outputs during training to improve the model performance; and (4) adopts the multi-task fusion loss to learn the different abilities for the models. It achieves a good performance on the Visual Question Answering (VQA) task, which shows the feasibility and effectiveness of the model. This solution can help blind or visually impaired individuals obtain visual information.

Keywords: vision-language models; multimodal learning; meta-learning; token-level training; visual question answering

MSC: 68T45



Citation: Ma, H.; Fan, B.; Ng, B.K.; Lam, C.-T. VL-Meta: Vision-Language Models for Multimodal Meta-Learning. *Mathematics* **2024**, *12*, 286. <https://doi.org/10.3390/math12020286>

Academic Editor: Jie Wen

Received: 29 December 2023

Revised: 12 January 2024

Accepted: 13 January 2024

Published: 16 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Asking a question about an image is a straightforward and common task for people, especially children learning things. Children strengthen their understanding of the world through continuous learning, and this transferable general knowledge can help them complete complex tasks, such as looking at pictures and answering questions. However, it is hard for an artificial intelligence (AI) model. It needs much data, computing power, and time because the large parameter models need to be fine-tuned by using large-scale downstream data to understand the task. AI can only learn tasks through datasets, and the relatively limited pre-training dataset is very limited compared to the general knowledge that children have. Moreover, children can complete complex tasks with only a few examples, which is also the goal we pursue in this work. Therefore, it is very meaningful to use a small amount of data to assist the model in completing the picture Visual Question Answering (VQA). It can also help visually impaired individuals perceive the environment.

Single-modal models only involve data from one modality during the training process, such as language models only learning language abilities and visual models learning visual abilities. Different modal data, like text, image, audio, video, etc., can complement each other to illustrate the same thing. However, different data have their own feature space, which cannot be directly used to output answers by the language decoder. How to align the different feature spaces to make the model understand different modal data, even with limited labeled data, is a challenging and valuable problem.

Multimodal learning can make models understand different modal data and combine different modal features. It uses different modal data to deal with a multimodal problem like the VQA task [1–9]. There are many works [10–14] trying to solve this problem. Some works [15–24] use relevant models to obtain the features from different modal data or, further, input them to a fusion module to combine different modal features to make the model understand the different modal data and finally perform the multimodal task. Some works [13,14,25,26] use only one encoder to make the model learn different multimodal data.

Meta-learning [27–38] can deal with the few-shot problem. It helps models to learn by constructing tasks with different support and query sets. The support set is used to train the model, and the query set is used to evaluate the model. It can set the N categories and K support samples to train an N -way K -shot meta-learning model. Then, the model can be applied to a new task.

To bridge the multimodal feature spaces gap with a few examples, Frozen [39] trains a vision encoder to embed each image as a vision prefix, which can help the language decoder understand images. But it needs much data and computing power to train the vision encoder, which is not friendly for low-source learners. MML [40] proposes a mapper module to convert the vision feature to language feature space with limited data. It can reduce the training computation, but the vision feature cannot learn the language information, which is important for multimodal tasks.

Our model VL-Meta is a simple but effective method, as shown in Figure 1. Our main contributions are as follows:

- We present the VL-Meta model structure, including the vision-language mapper and multimodal fusion mapper, simple and light networks that make the large language models understand the visual features by mapping the vision features into language feature space and fusing the final feature, as shown in Section 3.4.
- We propose the meta-task pool that constructs the support set and query set by meta-learning for training and validating the model, as shown in Section 3.2.
- We present the token-level training that makes the training phase align with the generation phase that can import the performance of the model, as shown in Section 3.3.
- We adopt the multi-task fusion loss to help the model learn from different perspectives of data and tasks, as shown in Section 3.5.

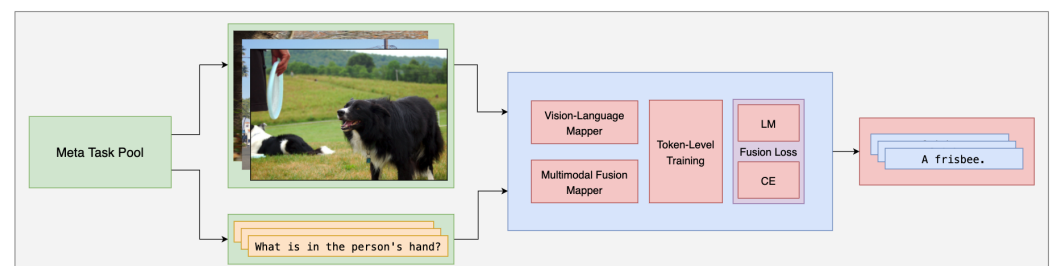


Figure 1. VL-Meta flow chart. The meta-task pool constructs the support set and query set from training data based on their classification to make the model learn the data knowledge and task knowledge. The vision-language mapper maps the vision feature into the language feature space, and the multimodal fusion mapper fuses the whole input features, which helps the model understand the images. The token-level training aligns the input with the output token by token. We fuse 2 losses to make the model learn data and tasks from different perspectives. Finally, the model can output the answer based on the images and the question.

The following sections are shown as follows. Section 2 introduces the related work. Section 3 describes the method details of VL-Meta. Section 4 introduces the data, model, and algorithm in the experiments. Section 5 summarizes the conclusions and some future work.

2. Related Work

Previous works that have tried to solve the multimodal problem are summarized below.

2.1. Multimodal Learning

Some multimodal data, including text, image, audio, video, code, table, and so on, are used to train models. However, traditional models can only deal with single-modal data. In this work, we focus on vision-language multimodal learning. Vision-language processing (VLP) focuses on image and text data and combines them into one new feature. Some representative studies that are worthy of attention are discussed as follows.

UNITER [15] uses vision and language encoders to extract the vision feature and the language feature and then combines them by using a transformer structure to fuse these two different modal inputs. It concatenates the two different types of modal features and then adopts four objects to train the model. Masked language modeling (MLM) learns the language-understanding ability by using the masked text. Masked region modeling (MRM) learns the vision-understanding ability by using the masked image region. Word region alignment (WRA) learns the local interaction information between images and text. Image-text matching (ITM) learns the multimodal alignment ability by using the text and image pair. Based on these four objects, UNITER uses the multimodal data to train the whole model.

ViLT [16] assumes that the multimodal data can give more information by the fusion inputs. It uses some extra embeddings such as *[class]* embedding, modal-type embedding, token position embedding, and patch position embedding. Using a *[class]* token to represent the whole text and image, each text and image has a *[class]* embedding to represent itself. Modal-type embedding represents the data modal. Here, 0 means text embedding, and 1 means image patch. The position embedding represents the text and image itself. There are three main loss functions to train the model. The MLM loss function trains the model to predict the masked token by using the remnant text embedding and image patches. The ITM loss function lets the model determine whether the text matches the image. The word patch alignment (WPA) loss function is similar to the above WRA. It calculates the distance between the text embedding and image embedding.

ALBEF [17] assumes that the text encoder does not need many multimodal layers but can obtain a good-enough feature representation. So, it splits the 12-layer transformer into a 6-layer text encoder and a 6-layer multimodal fusion module. It uses the image-text contrastive (ITC) loss function to obtain a better feature representation of image and text and uses ITM and MLM as the object functions to train the model. SimVLM [18] is an encoder-decoder architecture. It extracts the image and text embedding and concatenates them as the inputs to a transformer encoder-decoder module. The encoder can obtain the feature of the multimodal inputs, and the decoder can convert the multimodal feature to natural language to perform the downstream task. It adopts the PrefixLM loss, which inputs a sequence and generates the next sequence.

BLIP-2 [21] proposes a Q-former and introduces the learned query. The Q-former is the BERT transformer's structure that can align the vision feature to the language feature space. The learned query learns from different modal features, and it can control the length of the vision feature to reduce the computation. However, it cannot be improved by giving in-context VQA examples because the training set does not have this type of data. Ref. [41] proposes to adopt a visual transformer that aligns visual and linguistic features to enable the model to learn the local features of the image. Ref. [42] proposes inserting predefined length vectors to generate effective descriptions of input images, using the bird swarm algorithm (BSA) and long short-term memory (LSTM) models for sentence generation, to enhance image captioning performance. Ref. [43] proposes a method for constructing cross-modal graph convolutional networks for multimodal information fusion. Specifically, it introduces image titles as auxiliaries and aligns them with images to enhance semantic transmission. Then, it uses the generated sentences and images as nodes to construct the graph. Through graph learning, long-distance dependencies can be captured while filtering

visual noise. CLIPCap [44] also designs a light module to concatenate the vision and text features. It designs a mapping network to obtain the vision features and convert the shape into 10 lengths to make the language model understand the images.

2.2. Multimodal Meta-Learning

Meta-learning in the multimodal problem can improve the generalization performance with a few examples. Ref. [45] proposed to use meta-learning for low-resource data transfer in multimodal tasks. It adopts the large multimodal model to improve the cross-lingual performance of the model. Frozen [39] achieves the vision-language multimodal task by using the meta-learning dataset. It does not train the vision and language models but only trains a vision encoder to let the vision feature be understood by the language model. It also constructs a meta-learning dataset to make the model learn how to learn and perform a task that has not been seen before. MML [40] assumes that Frozen [39] needs to re-train the vision model, which requires much computing power and time. So, it designs a light module that can obtain the vision features by using four learnable prefix tokens and then extracts these four learnable prefix tokens as the image embedding and concatenates them with the text embedding. It also creates meta-datasets for training, validation, and testing. It does not re-train any pre-trained models but just trains a light module to make the language model understand the vision features.

3. VL-Meta

To train a model that can perform multimodal tasks with limited data under the meta-learning setting, we define the VQA task and the multimodal meta-learning setting, then formally introduce our contributions to the VL-Meta model structure, meta-task pool, token-level training, and multi-task fusion loss.

3.1. Problem Definition

There is a dataset D about Visual Question Answering (VQA) as Equations (1) and (2), which includes the I -th image (vision) V , the I -th question Q , and the I -th answer A .

$$D = \{(V, Q, A)\} \quad (1)$$

$$= \{(V_1, Q_1, A_1), \dots, (V_I, Q_I, A_I)\} \quad (2)$$

These three kinds of inputs are the sets that can be written as Equations (3)–(5). Each set has its elements, which are the i -th token v of the I -th image, the j -th token q of the I -th question, and the k -th token a of the I -th answer.

$$V_I = \{v_1^I, \dots, v_i^I\} \quad (3)$$

$$Q_I = \{q_1^I, \dots, q_j^I\} \quad (4)$$

$$A_I = \{a_1^I, \dots, a_k^I\} \quad (5)$$

To perform the VQA task, we propose *VL-Meta*, which can understand the images and answer the questions as Equations (6) and (7) where the I -th answer A is based on the I -th image V and the I -th question Q .

$$VL\text{-Meta}(V_I, Q_I) = A_I \quad (6)$$

$$VL\text{-Meta}(\{v_1, \dots, v_i\}, \{q_1, \dots, q_j\}) = \{a_1, \dots, a_k\} \quad (7)$$

3.2. Meta-Task Pool

For meta-learning, we first determine the N -way, which means the number of the image category, formulated as Equations (8) and (9). The name list includes all categories of the dataset. It randomly samples n time to construct the category list from the name list of the dataset.

$$category_list = random.sample(name_list, n) \tag{8}$$

$$= \{N_1, \dots, N_n\} \tag{9}$$

The support set can be constructed based on N -way K -shot as Equations (10) and (11) where the $(V, Q, A)_{K_k}^{N_n}$ is the k -th (V, Q, A) of the support set based on the n -th category, which can also be written as $K_k^{N_n}$.

$$support_set = \{(V, Q, A)_{K_1}^{N_1}, \dots, (V, Q, A)_{K_k}^{N_1}, \dots, (V, Q, A)_{K_1}^{N_n}, \dots, (V, Q, A)_{K_k}^{N_n}\} \tag{10}$$

$$= \{K_1^{N_1}, \dots, K_k^{N_1}, \dots, K_1^{N_n}, \dots, K_k^{N_n}\} \tag{11}$$

The query set can also be constructed as the support set based on N -way M -shot as Equations (12) and (13) where the $(V, Q, A)_{M_m}^{N_n}$ is the m -th (V, Q, A) of the query set based on the n -th category, which can also be written as $M_m^{N_n}$.

$$query_set = \{(V, Q, A)_{M_1}^{N_1}, \dots, (V, Q, A)_{M_m}^{N_1}, \dots, (V, Q, A)_{M_1}^{N_n}, \dots, (V, Q, A)_{M_m}^{N_n}\} \tag{12}$$

$$= \{M_1^{N_1}, \dots, M_m^{N_1}, \dots, M_1^{N_n}, \dots, M_m^{N_n}\} \tag{13}$$

To construct a meta-task, we combine the support set and the query set of each category N as Equations (14) and (15) where it concatenates the k -th (V, Q, A) of the support set and the m -th (V, Q, A) of the query set based on the n -th category.

$$meta_task = \{(V, Q, A)_{K_1}^{N_1}, \dots, (V, Q, A)_{K_k}^{N_1}, (V, Q, A)_{M_1}^{N_1}, \dots, (V, Q, A)_{M_m}^{N_1}, \dots, (V, Q, A)_{K_1}^{N_n}, \dots, (V, Q, A)_{K_k}^{N_n}, (V, Q, A)_{M_1}^{N_n}, \dots, (V, Q, A)_{M_m}^{N_n}\} \tag{14}$$

$$= \{K_1^{N_1}, \dots, K_k^{N_1}, M_1^{N_1}, \dots, M_m^{N_1}, \dots, K_1^{N_n}, \dots, K_k^{N_n}, M_1^{N_n}, \dots, M_m^{N_n}\} \tag{15}$$

Finally, based on Equations (1)–(15), the meta-task pool can repeat the meta-task T times to be constructed as Equations (16) and (17) where T_t is the t -th task T of the meta-task pool.

$$meta_task_pool = \{meta_task_1, \dots, meta_task_t\} \tag{16}$$

$$= \{T_1, \dots, T_t\} \tag{17}$$

The meta-task pool can adjust the task size t , category number n , support size k , and query size m to change its size, as shown in Figure 2.

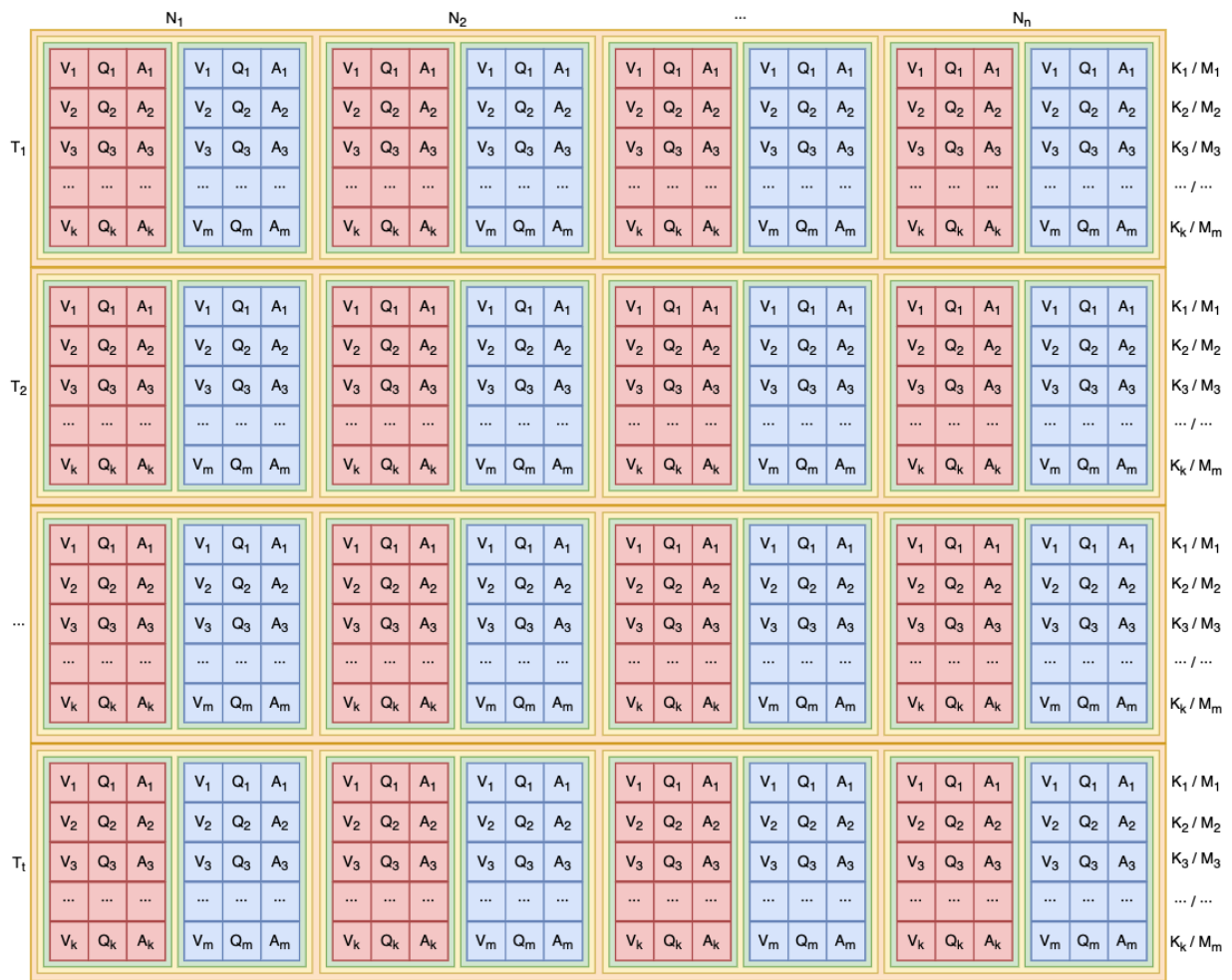


Figure 2. Meta-task pool can construct T tasks from training data. To construct a task, we first need to random sample N categories and then sample K sets of VQA for the support set and M sets of VQA for the query set based on each category.

We adjust the meta-task pool by using $T, N, K,$ and M . T means the number of tasks that adjust the size of the meta-pool task. N means the number of categories that adjust the size of the categories in a task. K means the size of the support set that includes K support set data for each category. M means the size of the query set that includes M pieces of query set data for each category. We propose the meta-task pool by meta-learning. We construct the support set to train the model and construct the query set to evaluate the model, for example: Category 1 of the support set is $[image_1, question_1, answer_1], [image_2, question_2, answer_2], [image_3, question_3, answer_3]$; Category 2 of the support set is $[image_1, question_1, answer_1], [image_2, question_2, answer_2], [image_3, question_3, answer_3]$; Category 1 of the query set is $[image_1, question_1, answer_1], [image_2, question_2, answer_2], [image_3, question_3, answer_3]$; Category 2 of the query set is $[image_1, question_1, answer_1], [image_2, question_2, answer_2], [image_3, question_3, answer_3]$; In a 2-way-3-shot setting, we train the model by the support set, including 2 categories and each category with 3 data, and evaluate the model by the query set, including 2 categories and each category with 3 data.

3.3. Token-Level Training

The token-level training is a method to align the input to the output token by token during training. There are many levels, such as document-level, paragraph-level, sentence-level, and token-level. For the token-level, it means that VL-Meta learns the training set into the token by token. Generative models such as the GPT series output token by

token, forming tokens and linking them into sentences. Therefore, we consider aligning the training method with this output method so that the input end can maintain the input-output form of this method, as shown in Figure 3.

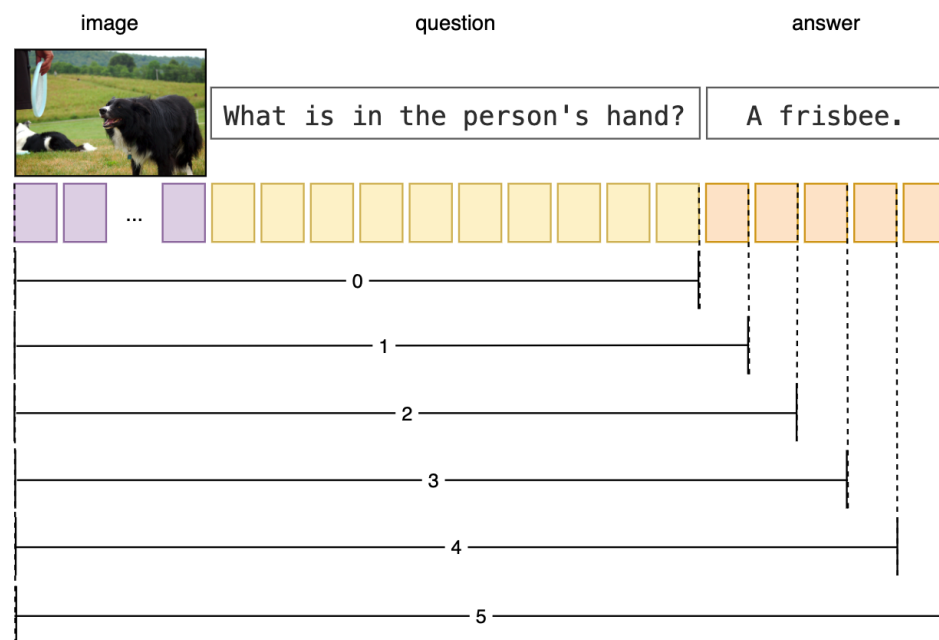


Figure 3. Token-level training is used to align the input to the output token by token. We set the input token as Position 0, which means there is no answer token in the input. Secondly, we add the first one of the answer tokens to the input as Position 1. Next, we move to the next position, Position 2, to give the model one more answer token, and so on. In this way, the model can learn the input tokens one by one to align to the output.

We proposed the method for this purpose, training token by token, that is, token-level training, or the token-for-token training method, which has more benefits than sentence-level training or sentence-for-sentence training. Therefore, we integrate this training method into the algorithm, which is called token-level training.

3.4. Model Structure

The vision-language mapper (VLM) inputs the vision feature and language features to map the images and questions to output the vision representation prompt, which includes these two multimodal data information to represent the images as Equations (18) and (19). The vision representation prompt VP token vp can provide the image information, and the length is l .

$$VLM(V, Q) = VP \tag{18}$$

$$= \{vp_1, \dots, vp_l\} \tag{19}$$

The multimodal fusion mapper (MFM) inputs the vision representation prompt VP and the question embeddings QE to map the whole inputs to the language feature space to make the language decoder understand the images and questions as Equations (20) and (21). The final inputs are (FV, FQ) with the token $\{vp_1, \dots, vp_l, qe_1, \dots, qe_m\}$.

$$MFM(VP, QE) = \{FV, FQ\} \tag{20}$$

$$= \{vp_1, \dots, vp_l, qe_1, \dots, qe_m\} \tag{21}$$

The visual features cannot be recognized by language models, but modal transformation can be achieved by training network structures vision-language mapper and multimodal

fusion mapper with only a small number of parameters in multimodal interaction, as shown in Figure 4.

VL-Meta is a simple but effective model. To extract the vision image features, we use the vision processor to embed the image into vision embedding vectors and then input the vision embedding vectors into the Frozen vision encoder to extract the vision image features. To extract the language question embedding vectors, we use the language tokenizer to embed the sentence into the language embedding vectors. The vision-language mapper network structure is a transformer layer that is used to map visual image features and language problem features into the vision representation alignment tokens. This network structure makes the generated new visual features more targeted to the problem and reduces the impact on the language model. The length of the token of sentences can be from 1 to 10 tokens. It can also be more than 10, but our method defines it as a mapper method, so the number can be smaller than the original sentences. The hidden size of the tokens is 768, which means the dimensionality of the embeddings and hidden states. The multimodal fusion mapper network structure is a linear layer that fuses new visual features, vision representation alignment, and language problem features, to stimulate the language ability of the language model, thus using a simple language model to complete visual-language tasks. Finally, we input all of them into the language decoder to generate the answer based on the image and the question.

The language model only used language data for training during the training process, without using data from other modalities. Therefore, after training, the language cannot recognize data from other models except for language. Our method VL-Meta uses vision-language mapper and multimodal fusion mapper network structures during the modal transformation process and sets task objectives during training. The former maps visual features to language space under the training of task objectives, The latter mapper under the training of the task objectives, maps the visual features to the language space and the problems corresponding to the images through a fully connected layer to make the visual image features consistent with the language problem features. Our method is a simple way to make the model learn new abilities without training the pre-trained models but only training light mappers that have fewer parameters than the pre-trained models. It uses mappers as modules in the model architecture that do not affect the pre-trained models and can easily change any other pre-trained models. The traditional method needs to train the language model, which has 163 M trainable parameters or the vision model, which has 87 M trainable parameters. However, the number of parameters of the vision-language mapper is only about 7 M, and the number of parameters of the multimodal fusion mapper is just about 0.59 M, which is less than the two pre-trained models and easy to achieve the objective.

3.5. Multi-Task Fusion Loss

To help the model learn common knowledge from multiple related tasks to improve model performance and generalization, we adopt the multi-task fusion loss. We adopt the language modeling (LM) task [46], which is the GPT series [47–49] models loss to predict the next token based on the previous tokens as Equation (22) where y_n means the last label token.

$$\mathcal{L}_{lm}(y_n) = - \sum_{n=1}^k \log P(y_n | x_1, \dots, x_m, y_1, \dots, y_{n-1}) \quad (22)$$

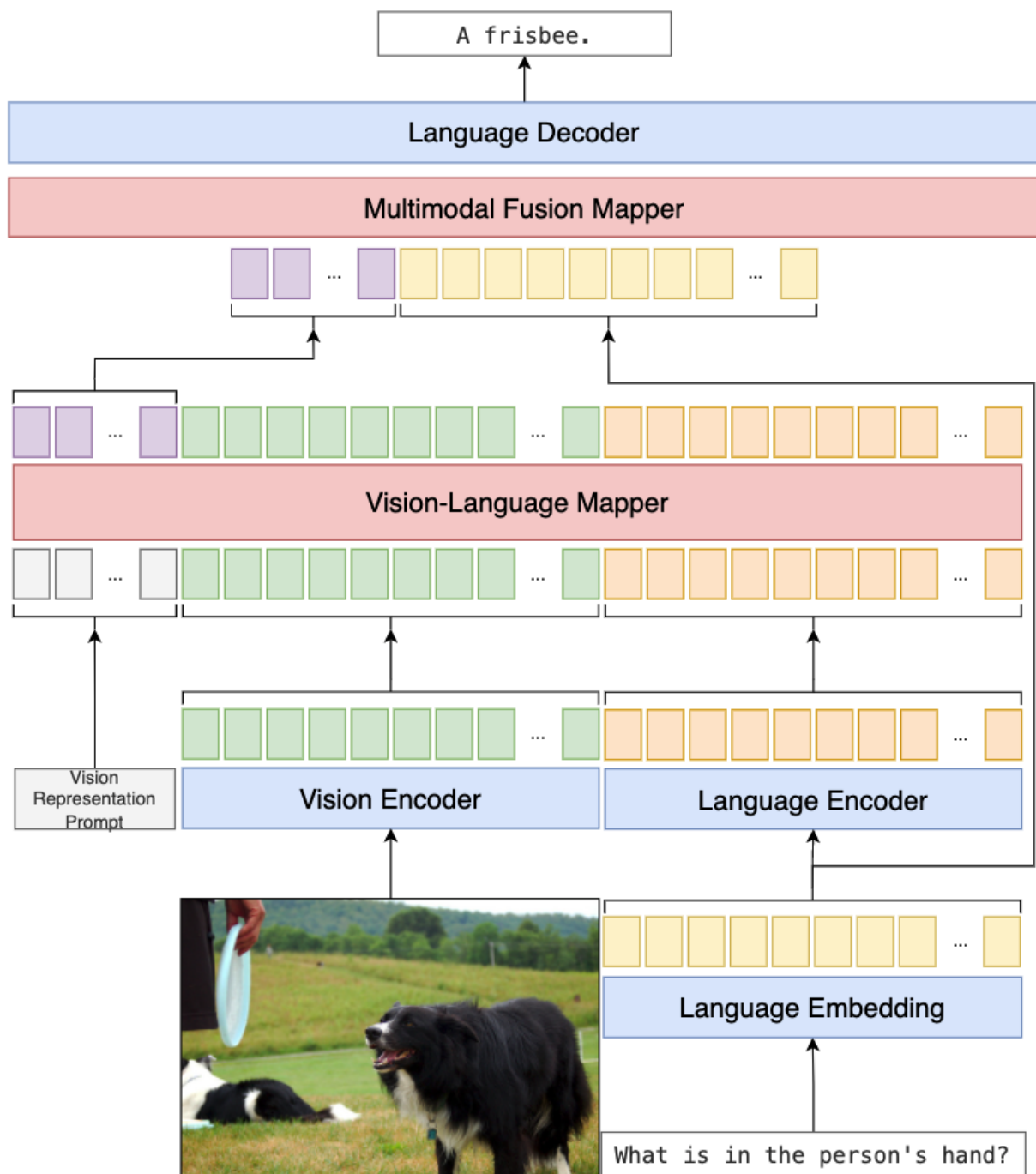


Figure 4. VL-Meta model architecture. First, we embed the question into the language embedding space by the language embedding layer to obtain the question embeddings, which are yellow blocks, and then encode the question embeddings into the language feature space by the language encoder to obtain the question features, which are the orange blocks. Second, to make the language decoder understand the images, we encode the images by the vision encoder to obtain the image features which are the green blocks and then map the vision feature and the question feature to the vision representation prompt by the vision-language mapper to represent the image with the corresponding to question information, which are the purple blocks. Third, we input the vision representation prompt and the question embeddings to the multimodal fusion mapper to obtain the final inputs. Fourth, we weigh the different losses to help the model learn all the abilities of the tasks. Finally, we input the inputs to the language decoder to obtain the answers for the VQA task.

We also use the cross-entropy (CE) task to calculate the prediction error during training as Equation (23), n is the index of the category from 1 to N . $\sum_{n=1}^N w_{y_n} \cdot \mathbb{1}\{y_n \neq \text{ignore_index}\}$ is a normalization term used to ensure that the weights of different samples are taken into account when calculating the average. The denominator of this term is the weight w_{y_n} for all samples multiplied by an indicator function. If y_n is not equal to ignore_index , then it is 1; otherwise, it is 0. This ensures that the weights of different samples are taken into account when calculating the average. w_{y_n} is the weight of category y_n used for weighting between different categories. This allows for the emphasis or reduction of certain categories of contributions in training. $\log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^C \exp(x_{n,c})}$ measures the error between the output probability distribution of the model and the actual label. $\{y_n \neq \text{ignore_index}\}$ is a means that specifies a target value like padding token that is ignored and does not contribute to the input gradient.

$$\mathcal{L}_{ce} = - \sum_{n=1}^N \frac{w_{y_n} \log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^C \exp(x_{n,c})} \cdot \mathbb{1}\{y_n \neq \text{ignore_index}\}}{\sum_{n=1}^N w_{y_n} \cdot \mathbb{1}\{y_n \neq \text{ignore_index}\}}. \tag{23}$$

Based on Equations (22) and (23), the final task loss is the weighted sum of the losses as Equation (24), w_{lm} is the weight of the LM loss, and w_{ce} is the weight of the CE loss.

$$\mathcal{L} = w_{lm} * \mathcal{L}_{lm} + w_{ce} * \mathcal{L}_{ce}. \tag{24}$$

The multi-task fusion loss is the \mathcal{L} which is our object to train the model to learn the vision understanding ability of the model.

3.6. Pseudocode

The pseudocode for the implementation of VL-Meta is shown in Algorithm 1, and it can be summarized as follows. First, load the training, evaluation, and test data from the COCO2017 and Guided VQA data sets. Second, load the pre-trained models to extract the features of images and texts. Third, train the model with a support set and evaluate the model with a query set. Specifically, in the training phase, the loss consists of two parts. They are language modeling loss and cross-entropy loss. Finally, test the model on the COCO2017 for intra-domain or Guide VQA dataset for cross-domain.

Algorithm 1 VL -Meta Pseudocode.

```

1: coco_data_list ← image_id, images, question, categories ← coco2017_preprocess(dataset)
2: coco_task_list ← support_set, query_set ← meta_pool(dataset = data_list, meta_task_pool_size = T, n, k, m)
3: guided_vqa_list ← load_test_data(dataset = guided_vqa)
4: vision_model = load_model.from_pretrained(vision_model_name)
5: language_model = load_model.from_pretrained(language_model_name)
6: for support_set, query_set in task_list do
7:   support_logits ← VL_Meta(support_set)
8:   lm_loss ← get_lm_loss(support_logits, support_labels)
9:   ce_loss ← get_ce_loss(support_logits, support_labels)
10:  loss ← lm_loss_weight * lm_loss + ce_loss_weight * ce_loss
11:  evaluation_logits ← VL_Meta(query_set)
12:  pred_token, get_pred_token(evaluation_logits)
13:  validation_acc ← torch.eq(pred_token, answer_token)
14: end for
15: for images, questions, answers in test_vqa_list do
16:  test_logits ← VL_Meta(images, questions)
17:  pred_token, get_pred_token(test_logits)
18:  test_acc ← torch.eq(pred_token, answer_token)
19: end for

```

4. Experiment

4.1. Datasets

The training and validation datasets are from the Microsoft COCO 2017 (MSCOCO2017) dataset [50], which is about image captioning. We first create a script for collecting and splitting the COCO datasets as training data and validation data. Specifically, each piece of data includes image ID, image, captions, and categories. The captions include five image captions. The categories are some classifications of the image objects. Then, we build a meta-learning set which is a task pool including some training and validation data. Further, the number of training and validation datasets can change depending on the task demand. As our experiments are under meta-learning, we randomly sample the training data and validation data from the meta-task pool. In the training period, we train the model by each token, which means the model will learn a sentence from the first word to the last word. It can give the model more opportunities to learn the sentence, especially paying attention to each word. That is also a way to the position IDs strategy. In the validation period, the model will generate tokens one by one. The accuracy is calculated by each token of prediction and ground truth. The test data are split into two data sets: one is intra-data, which is the same domain dataset as the train set, and the other is inter-data, which is another data set named Guided VQA. The cross-domain test dataset Guided VQA is constructed from the VisualGenome (VG) dataset [9], which includes some different tasks such as region descriptions, visual question answers, object instances, attributes, and relationships. It is visual-language representation learning.

Multimodal meta-learning constructs meta-tasks t , n categories k samples for training, and m samples for validation. The less n , more k , and the less m , the easier for training and validating. It can improve the model to perform a new domain task. Specifically, we use MSCOCO2017 as the training set and the intra-domain test set, and VG as the cross-domain test set. We determine the n -way k -shot based on the task objectives, and sample and train it in the constructed meta-learning task pool. Meta reconstructs the dataset and provides more category samples for the query through support sets, providing more category information for the model. The cross-domain test can test whether the model has generalization ability and whether the model can complete tasks beyond the training set.

4.2. Implementation Details

In this experiment, we just re-use the pre-trained models and freeze them to keep the parameters. Specifically, we use CLIP [15] with the vision transformer (ViT/B-32) [51] as a vision encoder to obtain the image features and use the GPT-2 [48] as a language encoder to obtain the text features. We tokenize the original text and image and then embed these tokens to vector space, and finally obtain the features of texts and images.

For the few-shot problem, we set the meta-task pool size as $T = 4$, which means the meta-task pool has 10 tasks; the category number as $N = 2$, which is to align with others; the support size in each task as $K = 1$ or $K = 5$ to test the support set performance in different settings; and the query size in each task as $M = 5$ to test the model performance in different settings. The total training data number is $T * N * K = 4 * 2 * 1 = 8$ or $T * N * K = 4 * 2 * 5 = 40$, and the total validating data number is $T * N * M = 4 * 2 * 5 = 40$.

The optimizer is AdamW [52], and the learning rate is $3e - 5$. To obtain a good initial learning rate, we adopt a warm-up strategy and choose the linear schedule with a warm up. During the warm-up period, the learning rate will gradually increase from 0 to the initial set. In the subsequent remaining training, the learning rate will gradually decrease from the initial set to 0. We adopt the early stop strategy on the validation set to avoid the model overfitting. We set the temperature as 0.1 to clearly distinguish between positive and negative samples.

4.3. Evaluation Metric

Accuracy is an important evaluation metric to measure the performance of the models. To easily compare the results, we use the accuracy as the same as other research, as Equation (25).

$$Accuracy = \sum_1^n \frac{torch.eq(pred_token_n, answer_token_n).sum().item()}{answer_token_n.numel()} \quad (25)$$

The molecule is the sum of the equal of the two tensors, the denominator is the number of one tensor element, and finally sum of each task from 1 to N .

4.4. Results and Discussion

Two models can compare the results because existing models, particularly not large ones, have fewer researchers to follow.

Frozen [39] proposes to freeze the language encoder and only train the vision encoder to make the model know the image and then perform the VQA task.

MML [40] presents to freeze the vision encoder and language encoder but design a light module called meta-mapper to train the model. Specifically, it concatenates a sequence of zero vectors called visual prefixes. The meta-mapper can convert the image feature to a visual prefix, which gives the language encoder the image information.

VL-Meta proposes to construct the meta-data, which can improve the generalization of the model; adopt the vision-language mapper, which can convert the vision feature and language feature to make the model understand the images; and propose token-level training, which can align outputs with inputs.

As shown in Algorithm 1, VL-Meta loads the support set and query set in the training set. VL-Meta learns the knowledge in the support set and uses the query set for validation. The model performance is evaluated through the test set, and the VQA task is ultimately completed.

As shown in Table 1, Frozen [39] only has the non-meta experimental result and is the baseline for multimodal few-shot learning. The MML [40] proposes to use meta-learning to improve the multimodal few-shot learning, which is more accurate than Frozen. The best accuracy in the real-fast VQA dataset is our VL-Meta, which reaches 10.936% for the two-way one-shot and 14.456% for two-way five-shots.

Table 1. VL-Meta comparison on 2-way cross-domain real-fast VQA task in accuracy. The bold font is the best accuracy in the results.

Methods	Accuracy (%)		
	Meta	1-Shot	5-Shots
Frozen [39]	×	7.8	10.5
MML [40]	×	6.9	10.7
	✓	8.5	13
VL-Meta (Ours)	×	7.952	10.864
	✓	10.936	14.456

For few-shot learning, Frozen trains on conceptual captions [53], a public dataset that consists of around three million image-caption pairs. We train the model using only 8 and 40 image and text pairs for $T * N * K = 4 * 2 * 1 = 8$ and $T * N * K = 4 * 2 * 5 = 40$, respectively. These quantities are equivalent to one three millionth and one seventy-five thousandth of the original data volume, which is consistent with MML.

For the model structure shown in Figure 4, we propose VL-Meta to use a Frozen vision encoder to extract vision image features and a Frozen language encoder to extract the language question features. Then, we input the vision image feature and language question feature to the vision-language mapper, which is just a transformer [54] in BERT [55], to

combine the two different modal features. Finally, we use the new feature as the vision prompt and input it with the question to generate the answer sentence. Note that the vision-language mapper is different from MML [40], which is without text features in training.

4.5. Ablation Study

In the ablation experiment, we test two dimensions of meta- and cross-domain. In Table 2, it shows the ablation experiment results. We found without the meta-learning, the model cannot perform better in the VQA task. In addition, cross-domain validation can prove that the model has good generalization ability. When increasing the support set from 1 to 5, all performances of the models can be improved.

Table 2. Ablation experiment of comparison of meta- and cross-domain of VL-Meta on 2-way real-fast VQA task in accuracy. The bold font is the best accuracy in the results.

Method	Meta	Cross-Domain	Accuracy (%)	
			1-Shot	5-Shots
VL-Meta	×	×	5.750	6.700
	×	✓	7.952	10.864
	✓	×	7.100	8.600
	✓	✓	10.936	14.456

As shown in Table 3, the accuracy is decreased without the meta-task pool and cross-domain. The performance is decreased when we adopt the linear layer as the vision-language mapper because the linear layer cannot learn much knowledge from the training data. Without the token-level training method, the model is unable to learn the generative features of each word in detail. The CE objective can help the model learn the differences between the prediction sentence and the ground truth answer, which can improve the training effect and the generation ability.

Table 3. Ablation experiment of comparison of VL-Meta on 2-way real-fast VQA task in accuracy. The bold font is the best accuracy in the results.

No.	Methods	Accuracy (%)	
		1-Shot	5-Shots
1	Meta-task pool (w/o)	7.952	10.864
2	Cross-domain (w/o)	7.100	8.600
3	Vision-language mapper (linear)	7.280	9.096
4	Token-level training (w/o)	8.304	9.496
5	Multi-task fusion loss (LM)	7.864	8.464
6	VL-Meta	10.936	14.456

4.6. Qualitative Analysis

Our VL-Meta can enable the language model to see images and improve the vision-understanding ability of the model to answer the questions based on the images, as shown in Table 1. It demonstrates that the model can use the vision prompt to answer the visual questions. However, for comparison with other works, we adopt accuracy as the metric that mechanically compares results instead of considering the semantic information. That can serve as one of the future directions.

5. Conclusions

It is generally difficult for a language model to reach good performance using a few examples, i.e., solving the few-shot problem. To alleviate this problem, we propose VL-Meta. First, we propose the vision-language mapper and the multimodal fusion mapper, which are light modules to make the language model able to learn the vision data like

images to perform tasks like VQA. Second, we propose the meta-task pool, which improves the model generalization to help the model learn both data knowledge and task knowledge. Third, we propose the train by each token method to align the inputs to outputs. Fourth, we adopt the multi-task fusion loss to achieve multiple tasks learning to let the model learn from different perspectives of the data and task. The experiments show that our method can reach a better performance. The results show that VL-Meta can extract vision and language features and can fusion them to perform multimodal learning.

Our research scenario is a few-shot problem, so the amount of data used is not significant. Therefore, during the evaluation process, it may lead to errors in the content of the answers. This problem may be due to the model not having training data to learn transferable knowledge, so one solution is to provide the model with more data to improve its ability to solve tasks. Our VL-Meta still needs to improve its performance to empower production and services in the industry, which is a limitation that needs to be a concern.

In the future, we consider some further research directions. The quality of data and the level of preprocessing can affect the effectiveness of model learning. This work used the same dataset as other models and maintained the same preprocessing method to explore whether this visual-language model is an effective method for achieving visual question-answering tasks. So, we can explore it as one of the future research directions. Training a model based on pre-trained models is an easy way to perform the multimodal task. Therefore, how to use the existing model to perform the multimodal task based on few-shot learning is an important future research direction.

Author Contributions: Conceptualization, H.M.; methodology, H.M.; software, H.M.; validation, H.M. and B.F.; formal analysis, H.M.; investigation, H.M. and B.F.; resources, B.K.N.; data curation, H.M. and B.F.; writing—original draft preparation, H.M.; writing—review and editing, H.M., B.F., B.K.N. and C.-T.L.; visualization, H.M. and B.F.; supervision, B.K.N. and C.-T.L.; project administration, B.K.N.; funding acquisition, B.K.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Macao Polytechnic University, grant number RP/ESCA-02/2021.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: <https://cocodataset.org/#download> (accessed on 24 June 2021), https://storage.googleapis.com/dm-few-shot-learning-benchmarks/guided_vqa.tar.gz (accessed on 3 November 2021).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ren, M.; Kiros, R.; Zemel, R. Exploring models and data for image question answering. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–12.
2. Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C.L.; Parikh, D. VQA: Visual Question Answering. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2425–2433.
3. Yu, L.; Park, E.; Berg, A.C.; Berg, T.L. Visual Madlibs: Fill in the blank Image Generation and Question Answering. *arXiv* **2015**, arXiv:1506.00278.
4. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
5. Johnson, J.; Hariharan, B.; Van Der Maaten, L.; Fei-Fei, L.; Lawrence Zitnick, C.; Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2901–2910.
6. Acharya, M.; Kafle, K.; Kanan, C. TallyQA: Answering complex counting questions. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 8076–8084. [[CrossRef](#)]
7. Shah, S.; Mishra, A.; Yadati, N.; Talukdar, P.P. KVQA: Knowledge-Aware Visual Question Answering. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 8876–8884. [[CrossRef](#)]
8. Andreas, J.; Rohrbach, M.; Darrell, T.; Klein, D. Neural module networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 39–48.

9. Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.J.; Shamma, D.A.; et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.* **2017**, *123*, 32–73. [[CrossRef](#)]
10. Jiang, H.; Misra, I.; Rohrbach, M.; Learned-Miller, E.; Chen, X. In defense of grid features for visual question answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10267–10276.
11. Wu, C.; Liu, J.; Wang, X.; Li, R. Differential Networks for Visual Question Answering. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 8997–9004. [[CrossRef](#)]
12. Fukui, A.; Park, D.H.; Yang, D.; Rohrbach, A.; Darrell, T.; Rohrbach, M. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv* **2016**, arXiv:1606.01847.
13. Li, X.; Yin, X.; Li, C.; Zhang, P.; Hu, X.; Zhang, L.; Wang, L.; Hu, H.; Dong, L.; Wei, F.; et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXX 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 121–137.
14. Su, W.; Zhu, X.; Cao, Y.; Li, B.; Lu, L.; Wei, F.; Dai, J. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. *arXiv* **2020**, arXiv:1908.08530.
15. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 8748–8763.
16. Kim, W.; Son, B.; Kim, I. Vilt: Vision-and-language transformer without convolution or region supervision. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 5583–5594.
17. Li, J.; Selvaraju, R.; Gotmare, A.; Joty, S.; Xiong, C.; Hoi, S.C.H. Align before fuse: Vision and language representation learning with momentum distillation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 9694–9705.
18. Wang, Z.; Yu, J.; Yu, A.W.; Dai, Z.; Tsvetkov, Y.; Cao, Y. SimVLM: Simple Visual Language Model Pretraining with Weak Supervision. *arXiv* **2022**, arXiv:2108.10904.
19. Chen, X.; Wang, X.; Changpinyo, S.; Piergiovanni, A.; Padlewski, P.; Salz, D.; Goodman, S.; Grycner, A.; Mustafa, B.; Beyer, L.; et al. PaLI: A Jointly-Scaled Multilingual Language-Image Model. *arXiv* **2023**, arXiv:2209.06794.
20. Wang, P.; Wang, S.; Lin, J.; Bai, S.; Zhou, X.; Zhou, J.; Wang, X.; Zhou, C. ONE-PEACE: Exploring One General Representation Model Toward Unlimited Modalities. *arXiv* **2023**, arXiv:2305.11172.
21. Li, J.; Li, D.; Savarese, S.; Hoi, S. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. *arXiv* **2023**, arXiv:2301.12597.
22. Yan, M.; Xu, H.; Li, C.; Tian, J.; Bi, B.; Wang, W.; Xu, X.; Zhang, J.; Huang, S.; Huang, F.; et al. Achieving Human Parity on Visual Question Answering. *ACM Trans. Inf. Syst.* **2023**, *41*, 1–40. [[CrossRef](#)]
23. Li, C.; Xu, H.; Tian, J.; Wang, W.; Yan, M.; Bi, B.; Ye, J.; Chen, H.; Xu, G.; Cao, Z.; et al. mPLUG: Effective and Efficient Vision-Language Learning by Cross-modal Skip-connections. *arXiv* **2022**, arXiv:2205.12005.
24. Yu, J.; Wang, Z.; Vasudevan, V.; Yeung, L.; Seyedhosseini, M.; Wu, Y. CoCa: Contrastive Captioners are Image-Text Foundation Models. *arXiv* **2022**, arXiv:2205.01917.
25. Wang, W.; Bao, H.; Dong, L.; Bjorck, J.; Peng, Z.; Liu, Q.; Aggarwal, K.; Mohammed, O.K.; Singhal, S.; Som, S.; et al. Image as a Foreign Language: BEiT Pretraining for All Vision and Vision-Language Tasks. *arXiv* **2022**, arXiv:2208.10442.
26. Bao, H.; Wang, W.; Dong, L.; Liu, Q.; Mohammed, O.K.; Aggarwal, K.; Som, S.; Piao, S.; Wei, F. VLMo: Unified vision-language pre-training with mixture-of-modality-experts. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 32897–32912.
27. Thrun, S.; Pratt, L. Learning to learn: Introduction and overview. In *Learning to Learn*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 3–17.
28. Vanschoren, J. Meta-Learning: A Survey. *arXiv* **2018**, arXiv:1810.03548.
29. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-learning in neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5149–5169. [[CrossRef](#)] [[PubMed](#)]
30. de Faria, A.C.A.M.; de Castro Bastos, F.; da Silva, J.V.N.A.; Fabris, V.L.; de Sousa Uchoa, V.; de Aguiar Neto, D.G.; dos Santos, C.F.G. Visual Question Answering: A Survey on Techniques and Common Trends in Recent Literature. *arXiv* **2023**, arXiv:2305.11033.
31. Wang, J.X. Meta-learning in natural and artificial intelligence. *Curr. Opin. Behav. Sci.* **2021**, *38*, 90–95. [[CrossRef](#)]
32. Rafiei, A.; Moore, R.; Jahromi, S.; Hajati, F.; Kamaleswaran, R. Meta-learning in healthcare: A survey. *arXiv* **2023**, arXiv:2308.02877.
33. Gharoun, H.; Momenifar, F.; Chen, F.; Gandomi, A.H. Meta-learning approaches for few-shot learning: A survey of recent advances. *arXiv* **2023**, arXiv:2303.07502.
34. Wang, C.; Zhu, Y.; Liu, H.; Zang, T.; Yu, J.; Tang, F. Deep Meta-learning in Recommendation Systems: A Survey. *arXiv* **2022**, arXiv:2206.04415.
35. yi Lee, H.; Li, S.W.; Vu, N.T. Meta Learning for Natural Language Processing: A Survey. *arXiv* **2022**, arXiv:2205.01500.
36. Mandal, D.; Medya, S.; Uzzi, B.; Aggarwal, C. Metalearning with graph neural networks: Methods and applications. *ACM SIGKDD Explor. Newsl.* **2022**, *23*, 13–22. [[CrossRef](#)]
37. Huisman, M.; Van Rijn, J.N.; Plaata, A. A survey of deep meta-learning. *Artif. Intell. Rev.* **2021**, *54*, 4483–4541. [[CrossRef](#)]
38. Peng, H. A Comprehensive Overview and Survey of Recent Advances in Meta-Learning. *arXiv* **2020**, arXiv:2004.11149.

39. Tsimpoukelli, M.; Menick, J.L.; Cabi, S.; Eslami, S.; Vinyals, O.; Hill, F. Multimodal few-shot learning with frozen language models. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 200–212.
40. Najdenkoska, I.; Zhen, X.; Worring, M. Meta Learning to Bridge Vision and Language Models for Multimodal Few-Shot Learning. *arXiv* **2023**, arXiv:2302.14794.
41. He, L.; Liu, S.; An, R.; Zhuo, Y.; Tao, J. An End-to-End Framework Based on Vision-Language Fusion for Remote Sensing Cross-Modal Text-Image Retrieval. *Mathematics* **2023**, *11*, 2279. [[CrossRef](#)]
42. Omri, M.; Abdel-Khalek, S.; Khalil, E.M.; Bouslimi, J.; Joshi, G.P. Modeling of Hyperparameter Tuned Deep Learning Model for Automated Image Captioning. *Mathematics* **2022**, *10*, 288. [[CrossRef](#)]
43. Zeng, D.; Chen, X.; Song, Z.; Xue, Y.; Cai, Q. Multimodal Interaction and Fused Graph Convolution Network for Sentiment Classification of Online Reviews. *Mathematics* **2023**, *11*, 2335. [[CrossRef](#)]
44. Mokady, R.; Hertz, A.; Bermano, A.H. ClipCap: CLIP Prefix for Image Captioning. *arXiv* **2021**, arXiv:2111.09734.
45. Hu, H.; Keller, F. Meta-Learning For Vision-and-Language Cross-lingual Transfer. *arXiv* **2023**, arXiv:2305.14843.
46. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*; Association for Computational Linguistics: New Orleans, Louisiana, 2018; pp. 2227–2237. [[CrossRef](#)]
47. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. *Improving Language Understanding by Generative pre-Training*; OpenAI: San Francisco, CA, USA, 2018.
48. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
49. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
50. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. *arXiv* **2014**, arXiv:1405.0312.
51. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations, Virtual*, 3–7 May 2021.
52. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2017**, arXiv:1711.05101.
53. Sharma, P.; Ding, N.; Goodman, S.; Soricut, R. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Gurevych, I., Miyao, Y., Eds.; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 2556–2565. [[CrossRef](#)]
54. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
55. Kenton, J.D.M.W.C.; Toutanova, L.K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; Volume 1*, p. 2.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.