

# Zero-Shot and Few-Shot Video Question Answering with Multi-Modal Prompts

Deniz Engin<sup>1</sup>      Yannis Avrithis<sup>2</sup>

<sup>1</sup>Inria, Univ Rennes, CNRS, IRISA

<sup>2</sup>Institute of Advanced Research in Artificial Intelligence (IARAI)

## Abstract

*Recent vision-language models are driven by large-scale pretrained models. However, adapting pretrained models on limited data presents challenges such as overfitting, catastrophic forgetting, and the cross-modal gap between vision and language. We introduce a parameter-efficient method to address these challenges, combining multimodal prompt learning and a transformer-based mapping network, while keeping the pretrained models frozen. Our experiments on several video question answering benchmarks demonstrate the superiority of our approach in terms of performance and parameter efficiency on both zero-shot and few-shot settings. Our code is available at <https://engindeniz.github.io/vitis>.*

## 1. Introduction

Recent vision-language models have shown remarkable progress, driven by transformer-based *large-scale pretrained models* [10, 39, 9, 38, 17, 45, 44]. These models have been incorporated into video understanding methods, including *video question answering (VideoQA)*, through multimodal fusion on *large-scale multimodal datasets* [41, 3, 60]. However, adapting pretrained models to video-language tasks on limited data is challenging. This is because of the gap between the visual and language modalities and, more importantly, because finetuning the entire model on limited data can lead to overfitting and forgetting previously acquired knowledge.

To address the gap between modalities, transformer-based mapping networks have been employed between frozen vision and language models [42, 16, 1]. These networks map visual features to an appropriate embedding space before they are given as input to the language models. To address overfitting, parameter-efficient adaptation methods have been explored, *e.g.*, *prompt learning* [35, 37, 36] and *adapter layers* [18] on frozen pretrained models. These approaches preserve the generalization of large-scale models while reducing the number of trainable parameters.

In this work, we investigate the adaptation of large-

scale visual-language models to VideoQA under scarcity of training data. Inspired by FrozenBiLM [57], we incorporate visual inputs to a frozen language model using lightweight learnable adapter layers. Beyond that, we introduce a novel *visual mapping network* that summarizes the video input while allowing for temporal interaction, inspired by [42, 20]. In addition, we introduce *multimodal prompt learning*, which diminishes the number of stored parameters when finetuning in the few-shot setting. We call our model *VideoQA with Multi-Modal Prompts (ViTiS)*.

We pretrain trainable parameters of ViTiS, *i.e.* *visual mapping network, adapter layers, visual and text prompts*, under the *masked language modeling (MLM)* objective on video-text pairs collected from the web, while the vision and language models are kept frozen. We evaluate ViTiS in the zero-shot and few-shot settings. For the latter, we finetune the model on downstream VideoQA tasks, using two approaches: (i) fine-tuning all trainable parameters, which are 8% of the total model parameters, (ii) fine-tuning only the prompts, which are 0.8% of all trainable parameters and a mere 0.06% of the total model parameters.

Our extensive experimental results on multiple open-ended VideoQA datasets demonstrate that ViTiS outperforms prior methods while requiring fine-tuning of only a few parameters for each dataset in few-shot settings. In addition, our visual mapping network contributes to better alignment and understanding of multimodal inputs, improving performance in both zero-shot and few-shot settings.

Our contributions can be summarized as follows:

1. We introduce *multimodal prompt learning* to few-shot VideoQA for the first time, fine-tuning as low as 0.06% of model parameters on downstream tasks.
2. We introduce a *visual mapping network* to VideoQA, mapping video input to the text embedding space, while supporting temporal interaction.
3. We experimentally demonstrate the strong performance of ViTiS on multiple VideoQA datasets in both zero-shot and few-shot settings.

## 2. Related Work

**Video question answering** Recent advances in vision-language models benefit from pretrained foundation models, including vision-only [10, 39] language-only [9, 38, 17, 45] and vision-language [44]. Recent video understanding methods, including VideoQA, incorporate these models by leveraging large-scale multimodal data [41, 3, 60] with different pretraining objectives, *e.g.*, *masked language modeling*, *masked image modeling*, or *predicting the next word*, to perform single or multiple vision-language tasks [48, 33, 28, 12, 55, 60, 31, 57, 1, 59, 8, 51, 34, 19, 13].

Adapting pretrained vision-language models to downstream tasks relies on fully supervised fine-tuning on VideoQA datasets in general [50, 53, 21, 29, 58, 33, 14]. Few recent works address the challenge of limited data by focusing on zero-shot [55, 56, 57, 1, 59, 32, 34] and few-shot [57, 1] open-ended VideoQA tasks. Our work is similar to [57] in leveraging a frozen video encoder and language model with adapter layers. Beyond that, we introduce a transformer-based visual mapping network between the two models, allowing for temporal interaction. In addition, we incorporate multimodal prompt learning, allowing for efficient fine-tuning in few-shot settings.

**Parameter-efficient training** As the size of large-scale pretrained models grows, adapting them efficiently on limited data without overfitting in an emerging research problem. A common solution is *adapters*, introduced by [18] and employed for vision-language tasks [11, 57, 49].

Another common solution is *prompting*, referring to inserting tokens to the input to guide pretrained models on downstream tasks. Prompts can be handcrafted (discrete) [4] or learned (continuous vectors) [35]. Pretrained language models demonstrate remarkable generalization to zero-shot settings with handcrafted prompts [4]. Prompt learning is introduced initially in natural language processing tasks [35, 30, 37, 36, 43, 40] and subsequently adopted in vision [22, 2] and vision-language models. In the latter case, prompts are introduced to text encoders [62, 61], or both text and vision encoders [24, 52, 27, 46], called *multimodal*. Learnable prompts can be inserted at the input level [35] and/or deep layers [36, 22]. Few recent works employ prompt learning for video understanding [23, 63, 49] and multimodal prompt learning for video classification [52, 46]. We introduce multimodal prompt learning to few-shot VideoQA for the first time.

## 3. Method

The proposed method, ViTiS, is illustrated in Figure 1(a), consisting of a frozen video encoder, a visual mapping network, a frozen text embedding layer and a frozen language model that includes learnable text prompts and adapter layers. Given an input video  $X^v$ , represented as

a sequence of frames, and a question  $X^q$ , represented as a sequence of tokens, the problem is to predict an answer  $X^a$  that is another sequence of tokens. The model takes the concatenated sequence  $X^t = (X^q, X^a)$  as input text. Parts of  $X^t$  may be masked, for example  $X^a$  is masked at inference.

**Video encoder** The input video is represented by a sequence of  $T$  frames,  $X^v = (x_1^v, \dots, x_T^v)$ . This sequence is encoded into the *frame features*

$$Y^v := f^v(X^v) = (y_1^v, \dots, y_T^v) \in \mathbb{R}^{D \times T} \quad (1)$$

by a frozen pretrained *video encoder*  $f^v$ , where  $D$  is the embedding dimension.

**Visual mapping network** A *visual mapping network*  $f^m$  maps the frame features  $Y^v$  to the same space as the text embeddings. The mapping is facilitated by a set of  $M$  *learnable visual prompts*  $P^v \in \mathbb{R}^{D \times M}$ , which are given as input along with  $Y^v$ , to obtain the *video embeddings*

$$Z^v := f^m(P^v, Y^v) \in \mathbb{R}^{D \times M}. \quad (2)$$

As shown in Figure 1(c), the architecture of  $f^m$  is based on Perceiver [20], where the latent array corresponds to our learnable visual prompts  $P^v$ . It consists of  $L$  blocks defined as

$$Z_\ell := \text{SA}_\ell(\text{CA}_\ell(Z_{\ell-1}, Y^v)) \in \mathbb{R}^{D \times M} \quad (3)$$

for  $\ell = 1, \dots, L$ , with input  $Z_0 = P^v$ . Each block  $\ell$  maps the latent vectors  $Z_{\ell-1}$  first by cross attention  $\text{CA}_\ell$  with the frame features  $Y^v$  and then by self-attention  $\text{SA}_\ell$  to obtain  $Z_\ell$ . In cross attention,  $Z_{\ell-1}$  serves as query and  $Y^v$  as key and value. We thus iteratively extract information from the frame features  $Y^v$  into the latent vectors, which are initialized by the visual prompts. The output video embeddings are  $Z^v = Z_L \in \mathbb{R}^{D \times M}$ . To allow modeling of temporal relations within the video, learnable *temporal position embeddings* are added to  $Y^v$  before  $f^m$ .

**Text embedding** The input text is tokenized into a sequence of  $S$  tokens,  $X^t = (x_1^t, \dots, x_S^t)$ . This sequence is mapped by a frozen *text embedding layer*  $f^t$  to the text embedding space,

$$Z^t := f^t(X^t) = (z_1^t, \dots, z_S^t) \in \mathbb{R}^{D \times S}. \quad (4)$$

One or more tokens are masked, in which case they are replaced by a learnable mask token.

**Language model** We concatenate video and text embeddings into a single input sequence  $(Z^v, Z^t) \in \mathbb{R}^{D \times K}$  of length  $K = M + S$ . We then feed this sequence to a transformer-based bidirectional language model  $f$  to obtain an output sequence

$$f(Z^v, Z^t) \in \mathbb{R}^{D \times K} \quad (5)$$

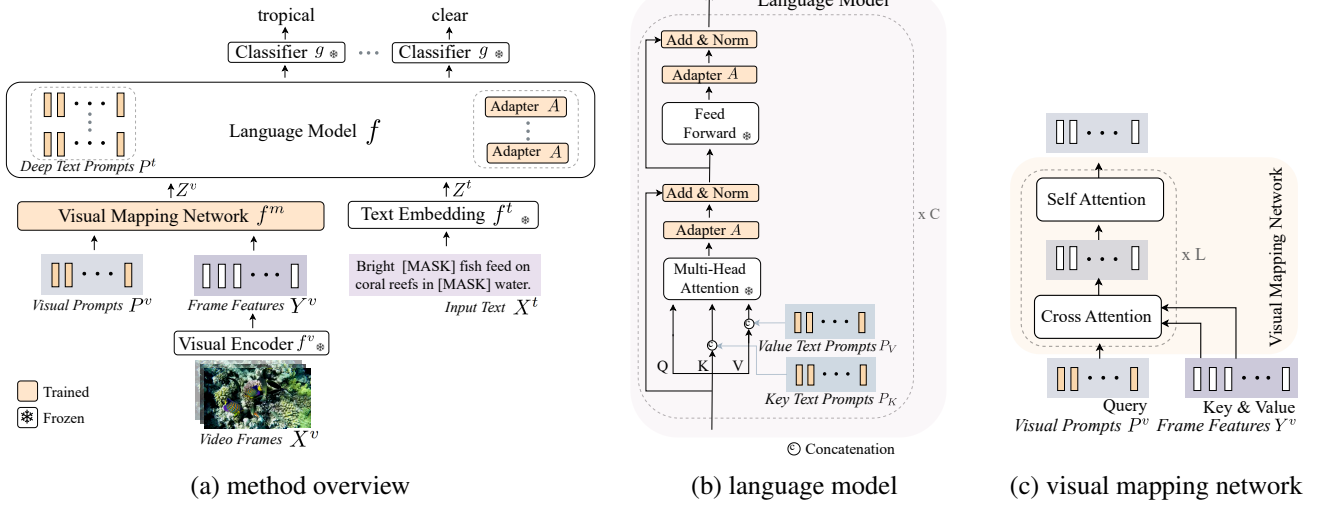


Figure 1: (a) ViTiS consists of a frozen video encoder  $f^v$ , a visual mapping network  $f^m$ , a frozen text embedding layer  $f^t$ , a frozen language model  $f$  and a frozen classifier head  $g$ . Given input video frames  $X^v$  and text  $X^t$ ,  $f^v$  extracts frame features and  $f^m$  maps them to the same space as the text embeddings obtained by  $f^t$ . Then,  $f$  takes the video and text embeddings  $Z^v$ ,  $Z^t$  as input and predicts the masked input tokens. (b) The *language model* incorporates learnable text prompts in the key and value of multi-head-attention and adapter layers after each self-attention and feed-forward layer, before LayerNorm. (c) Our *visual mapping network* consists of a number of layers, each performing cross-attention between learnable visual prompts and video frame features followed by self-attention.

of the same length. Finally, a classifier head  $g$  maps the output sequence to logit vectors over a vocabulary  $U$ . The logit vectors corresponding to masked tokens are selected to apply the loss function at training or make predictions at inference. Both  $f$  and  $g$  are pretrained and kept frozen. However, as shown in Figure 1(b),  $f$  is adapted by means of learnable deep text prompts and adapters, described next.

**Text prompts** To reduce the number of fine-tuned parameters at downstream tasks, we introduce attention-level text prompts in self-attention blocks at each layer of the language model, referred to as *deep text prompt learning* [36]. Given a sequence  $Z \in \mathbb{R}^{D \times K}$  of token embeddings as input to a self-attention layer of the language model  $f$ , we prepend two sequences of *learnable text prompts*  $P_K, P_V \in \mathbb{R}^{N \times D}$  to the key and value respectively:

$$Q := W_Q Z \quad K := [P_K \ W_K Z] \quad V := [P_V \ W_V Z], \quad (6)$$

where  $W_Q, W_K, W_V \in \mathbb{R}^{D \times D}$  are the query, key and value projections respectively. The output sequence length does not change since it is determined by the query, where we do not prepend prompts. There is one pair of variables  $P_K, P_V$  for each layer of  $f$ , collectively denoted as  $P^t$ . These variables are either defined as parameters directly or parametrized by means of projections as discussed next.

**Text prompt parametrization** Instead of defining text prompts as parameters directly, we discuss here an alternative parametrization using projections. We first generate

a sequence of input prompts  $P^i \in \mathbb{R}^{D' \times N}$  and then we project it as follows:

$$P^t := W P^i \in \mathbb{R}^{2CD \times N}, \quad (7)$$

where  $W \in \mathbb{R}^{2CD \times D'}$ ,  $C$  is the number of layers of the language model  $f$  and  $D$  its embedding dimension. Then,  $P^t$  can be reshaped as a  $2 \times C \times D \times N$  tensor, representing one pair of sequences  $P_K, P_V \in \mathbb{R}^{D \times N}$  for every layer of  $f$ . After training, the input sequence  $P^i$  and projection matrix  $W$  are discarded and we only keep  $P^t$ . This allows us to fine-tune fewer parameters at downstream tasks, which is beneficial when data is limited.

**Adapters** Following [57], we add adapter layers to the language model  $f$ . Given a sequence  $Z \in \mathbb{R}^{D \times K}$  of token embeddings, an adapter layer  $A$  maps it through a bottleneck dimension  $d$  with a residual connection:

$$A(Z) := Z + W_2 h(W_1 Z) \in \mathbb{R}^{D \times K}, \quad (8)$$

where  $W_1 \in \mathbb{R}^{d \times D}$ ,  $W_2 \in \mathbb{R}^{D \times d}$ , and  $h$  is the relu activation function. We insert an adapter module after the self-attention layer and the feed-forward layer, preceding LayerNorm in each layer of  $f$ .

**Training and inference** Our model is trained using the *masked language modeling* (MLM) objective, where one or more tokens of  $X^t$  are masked and the corresponding outputs are predicted over a vocabulary  $U$ . The parameters of the visual encoder  $f^v$ , text embedding layer  $f^t$ ,

language model  $f$  and classifier head  $g$  are pretrained and kept frozen. Only the newly introduced parameters, that is, visual prompts  $P^v$ , visual mapping network  $f^v$ , text prompts  $P^t$  and adapter layers, are optimized on video-text pairs. We then fine-tune these parameters on a smaller subset on downstream video question answering tasks, where  $X^t = (X^q, X^a)$  consists of a question-answer pair and masking applies to the  $X^a$  only. At inference,  $X^a$  is masked and the corresponding output yields a prediction.

## 4. Experiments

### 4.1. Datasets

**Pretraining** We use WebVid2M [3] for pretraining, consisting of 2.5M video-caption pairs scraped from the internet. The domain is open and the captions are manually generated. The average video duration is 18 seconds and the average caption word count is 12.

**Downstream tasks** Downstream dataset statistics are given in Table 1. Following [57], we use 1% of the training data for fine-tuning in the few-shot setting.

MSRVTT-QA [53] is an extension of MSR-VTT [54], where question-answer pairs are automatically generated from video descriptions. MSVD-QA [53] is based on MSVD [7] and question-answers pairs are automatically generated as in MSRVTT-QA. ActivityNet-QA [58] is derived from ActivityNet [6]. The average video duration is 180s. TGIF-QA [21] comprises several tasks, including FRAME-QA, where the question can be answered from one of the frames in a GIF. In this work, TGIF-QA refers only to Frame-QA.

### 4.2. Implementation details

**Architecture details** The *frozen video encoder* is CLIP ViT-L/14 [10, 44], trained with contrastive loss on 400M image-text pairs. We uniformly sample  $T = 10$  frames located at least 1 second apart and each frame is resized to  $224 \times 224$  pixels; if the video is shorter than 10 seconds, we zero-pad up to  $T = 10$  frames. The encoder then extracts one feature vector per frame of the dimension of 768, followed by a linear projection to  $D = 1536$  dimensions.

The *visual mapping network* has  $L = 2$  layers, each with a cross-attention and a self-attention, having 8 heads and embedding dimension  $D = 1536$ . We use  $M = 10$  learnable visual prompt vectors of dimension  $D = 1536$ .

The *text tokenizer* is based on SentencePiece [26] with a vocabulary  $U$  of size 128k.

The *frozen language model* is DeBERTa-V2-XLarge [17], trained using MLM on 160G text data, following [57]. The model has  $C = 24$  layers, 24 attention heads, and embedding dimension  $D = 1536$ , resulting in 900M parameters.

DATASET	VIDEOS	QA PAIRS			
		TRAIN	VAL	TEST	TOTAL
MSRVTT-QA [53]	10k	159k	12k	73k	244k
MSVD-QA [53]	2k	31k	6.5k	13k	50.5k
ActivityNet-QA [58]	5.8k	32k	18k	8k	58k
TGIF-QA [21]	40k	39k	-	13k	53k

Table 1: Downstream dataset statistics.

For the *adapter layers* [18], we set  $d = D/8 = 192$  by following [57]. For *text prompts*, we use  $N = 10$  learnable text prompt vectors,  $D' = D/8 = 192$ , and  $C = 24$ .

**Downstream input design** We limit the length of text sequences to  $S = 256$  tokens for pretraining and zero-shot experiments and  $S = 128$  tokens for downstream experiments. We adopt the input design of [57] as follows: "[CLS] Question: <Question>? Answer: [MASK]. Subtitles: <Subtitles> [SEP]". Subtitles are optional and if available, their token sequence  $X^s$  is incorporated into the input. In this case, the text input sequence becomes  $X^t = (X^q, X^a, X^s)$ .

**Answer vocabulary** The answer vocabulary  $U$  is constructed by selecting the top 1k most frequent answers from the training set for the zero-shot setting, following [57, 60]. Another vocabulary is formed by including answers that occur at least twice in the training set for the few-shot setting, as defined in [57]. Questions with answers outside the vocabulary are excluded from the training process and are assessed as incorrect during evaluation. To report results for the few-shot setting, we choose the vocabulary that yields the best performance on the validation set.

**Answer embedding** The classifier head of the frozen language model includes more tokens than required for downstream training. To address this, by following [57], we define a task-specific classification head by keeping the weights of the pretrained head associated with the answer vocabulary. At inference, we provide one mask token at the input, regardless of the ground truth answer length, and we obtain one output logit vector. For multi-token answers, we take the average of the logits corresponding to the ground truth words from the vocabulary.

**Evaluation Metrics** We report top-1 accuracy on public test sets for all downstream tasks, except TGIF-QA where we report on the validation set unless otherwise specified.

**Training settings** We use the Adam optimizer [25] with  $\beta = (0.9, 0.95)$  in all experiments. We decay the learning rate using a linear schedule with the warm-up in the first 10% of the iterations. We use dropout with probability 0.1 in the language model, adapter layers, text prompts, and visual mapping network. We adopt automatic mixed precision training for all experiments.

#	AD	MAP	PR	TRAINED PARAM	MSRVTT -QA	MSVD -QA	ANET -QA	TGIF -QA
1		Linear		1M	18.0	30.5	27.1	44.4
2		Linear	✓	15M	36.3	46.2	32.7	54.3
3	✓	Linear		30M	35.0	45.0	32.4	53.9
4	✓	Linear	✓	44M	36.4	47.2	32.9	54.7
5		VPN		58M	24.5	37.0	26.1	50.1
6		VPN	✓	72M	36.1	47.4	34.1	55.8
7	✓	VPN		86M	34.7	46.0	32.4	54.4
8	✓	VPN	✓	101M	<b>36.5</b>	<b>47.8</b>	<b>37.2</b>	<b>55.9</b>

Table 2: Effect of our proposed components on few-shot top-1 accuracy on the validation set. Pretraining on WebVid2M [3] followed by fine-tuning all trainable parameters on downstream datasets, using 1% of training data. AD: Adapters; MAP: mapping network; PR: text prompts; VPN: our visual mapping network. ANET-QA: ActivityNet-QA.

We *pretrain* for 10 epochs on WebVid2M with a batch size of 128 on 8 NVIDIA Tesla V100 GPUs, amounting to 20 hours total training time. The base learning rate is  $2 \times 10^{-5}$  and the learning rate for visual and text prompts is separately set to  $10^{-3}$ .

For *fine-tuning* on each downstream dataset, we train for 20 epochs with a batch size of 32 on 4 NVIDIA Tesla V100 GPUs. The base learning rate is searched over 5 values in the interval  $[10^{-5}, 5 \times 10^{-5}]$ , while the learning rate for visual and text prompts is kept at  $10^{-3}$ . For *prompt-only fine-tuning*, the base learning rate is searched over 3 values in the interval  $[10^{-3}, 3 \times 10^{-3}]$ .

### 4.3. Ablation

We conduct an ablation study in the few-shot setting.

**Model design** In Table 2, we analyze the effect of different components in the model design. We observe that changing the baseline from a linear layer to *our visual mapping network* without adapters increases the performance by a large margin in most datasets (row 1→5). By adding *text prompts* to any model design (row 1→2, 3→4, 5→6, 7→8), the performance increases for all datasets. The improvement is vast in the absence of adapters.

The model design that includes a linear mapping network and adapter layers (row 3) corresponds to FrozenBiLM [57] trained on WebVid2M. While using only our visual mapping network and text prompts (row 6) already works better than FrozenBiLM trained on WebVid2M, we further improve performance by incorporating adapter layers: our full model (row 8) achieves the best performance overall.

**Prompt length** Figure 2 shows the effect of the number of prompts on few-shot performance, referring to both visual ( $M$ ) and text ( $N$ ) prompts, *i.e.*,  $M = N$ . Because the space and time complexity of the model is quadratic in the number of prompts, we limit this number to 50. We find that

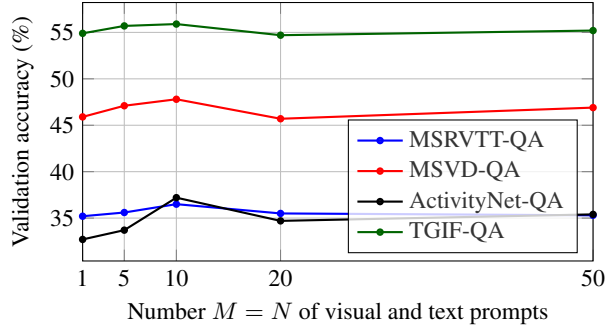


Figure 2: Few-shot top-1 validation accuracy vs. number  $M = N$  of *visual and text prompts* for different downstream datasets, using 1% of training data.

VPN LAYERS	MSRVTT -QA	MSVD -QA	ANET -QA	TGIF -QA
1	36.0	47.0	36.1	55.9
2	<b>36.5</b>	<b>47.8</b>	<b>37.2</b>	<b>55.9</b>

Table 3: Effect of number  $L$  of layers of our visual mapping network on few-shot top-1 validation accuracy, using 1% of training data. VPN: Visual Mapping Network. ANET-QA: ActivityNet-QA.

REPARAM	MSRVTT -QA	MSVD -QA	ANET -QA	TGIF -QA
	35.6	47.4	34.0	55.1
✓	<b>36.5</b>	<b>47.8</b>	<b>37.2</b>	<b>55.9</b>

Table 4: Effect of reparametrization of text prompts on few-shot top-1 validation accuracy, using 1% of training data. REPARAM: Reparametrization. ANET-QA: ActivityNet-QA.

accuracy is consistently best on all downstream benchmarks for  $M = N = 10$  prompts, which we choose as default.

**Number of layers of visual mapping network** Table 3 shows the effect of the number  $L$  of layers of our visual mapping network on few-shot performance. We only experiment with up to 2 layers to avoid an excessive number of parameters and complexity of our model. We find that  $L = 2$  works best, which we choose as default.

**Reparametrization of text prompts** In Table 4, we investigate the impact of the reparametrization of text prompts, as discussed in Subsection 4.2, on few-shot performance. We find that reparametrization consistently improves performance on all downstream benchmarks. Even though the number of trainable parameters increases from 87M to 101M during pretraining and fine-tuning, we do not need to store the additional parameters at inference.

#	INPUT DESIGN	MSRVTT -QA	MSVD -QA	ANET -QA	TGIF -QA
1	“[CLS] <Question>? [MASK]. <Subtitles> [SEP]”	13.2	30.2	19.8	29.8
2	“[CLS] <b>Answer the question:</b> <Question>? [MASK]. <Subtitles> [SEP]”	7.8	22.3	14.3	35.3
3	“[CLS] <Question>? <b>Answer:</b> [MASK]. <Subtitles> [SEP]”	17.7	37.2	<b>25.8</b>	45.1
4	“[CLS] <b>Question:</b> <Question>? <b>Answer:</b> [MASK]. <b>Subtitles:</b> <Subtitles> [SEP]”	<b>18.0</b>	<b>38.2</b>	24.9	<b>45.5</b>

Table 5: Effect of handcrafted prompt placement on *zero-shot* top-1 validation accuracy. ANET-QA: ActivityNet-QA.

#	INPUT DESIGN	MSRVTT -QA	MSVD -QA	ANET -QA	TGIF -QA
1	“[CLS] <Question>? [MASK]. <Subtitles> [SEP]”	36.3	47.0	35.8	55.8
2	“[CLS] <b>Answer the question:</b> <Question>? [MASK]. <Subtitles> [SEP]”	36.3	46.8	35.1	55.8
3	“[CLS] <Question>? <b>Answer:</b> [MASK]. <Subtitles> [SEP]”	<b>36.5</b>	47.1	35.9	55.8
4	“[CLS] <b>Question:</b> <Question>? <b>Answer:</b> [MASK]. <b>Subtitles:</b> <Subtitles> [SEP]”	<b>36.5</b>	<b>47.8</b>	<b>37.2</b>	<b>55.9</b>

Table 6: Effect of handcrafted prompt placement on *few-shot* top-1 validation accuracy, using 1% of training data. ANET-QA: ActivityNet-QA.

**Handcrafted prompts** We explore the use of handcrafted prompts in the input text. In Table 5 and Table 6, we consider four different input designs for zero-shot and few-shot settings, respectively: (i) no handcrafted prompts, (ii) placed before the question, (iii) placed just before the [MASK] token (answer), and (iv) placed just before the question, answer and subtitles.

In *zero-shot*, handcrafted prompts are beneficial due to the absence of task-specific learning for downstream tasks. As shown in Table 5, the absence of handcrafted prompts drastically reduces the performance (row 1), highlighting their necessity. Moreover, the position of the handcrafted prompt has a significant impact on the performance. More specifically, the location of the “Answer” prompt affects the results by a large margin (row 2→3), even leading to worse performance than the absence of handcrafted prompts (row 1→2). The presence of an “Answer” prompt just before the [MASK] token yields better performance in two input designs (rows 3 & 4).

Although the impact of using handcrafted text prompts is relatively small in *few-shot* experiments compared to zero-shot experiments, they still improve enhances, particularly on MSRVTT-QA and TGIF-QA datasets, as shown in Table 6. Placing handcrafted prompts at the beginning (row 2), as is the case for learnable text prompts, leads to lower performance. The best performance is achieved when handcrafted prompts are placed just before the question, answer, and subtitles (row 4). Therefore, we choose to place handcrafted prompts according to row 4 for both settings.

By contrast, *learnable prompts* are all placed at the beginning. We empirically observe that other choices, *e.g.* placing half at the beginning of the input and half just before the [MASK] token, are inferior.

## 4.4. Results

**Zero-shot** A comparison with state-of-the-art methods on open-ended zero-shot VideoQA is given in Table 7. We observe an outstanding performance of our method across different VideoQA benchmarks, despite using significantly less pretraining data compared to other methods. Our performance on ActivityNetQA [58] is on par with Frozen-BiLM [57]. Lavender [34] employs a multi-task training approach, transforming different vision-language tasks into MLM. Reserve [59] uses GPT-3 [5] to convert questions into masked sentences. Flamingo [1] uses a frozen auto-regressive language model trained on an extreme-scale dataset. By contrast, our method leverages a lighter frozen language model trained on 2.5M video-text pairs.

BLIP [32] is pretrained on the VQA dataset [15], which is not directly comparable as our setting does not involve training on QA pairs. Similarly, Just Ask [55, 56] uses automatically generated visual question answering datasets. Although these datasets are not annotated by humans, the model is still trained on the specific task.

**Few-shot** We fine-tune our method on 1% of the training data by following [57], which introduced the few-shot VideoQA task in this form. Table 9 compares our method with [57]. We use two strategies, fine-tuning (i) all trainable parameters and (ii) only prompts. The latter works best, consistently outperforming [57] while diminishing the number of fine-tuned parameters.

**Few-shot in-context learning** An alternative approach for few-shot VideoQA is *in-context learning* [1], using few, *e.g.* 32, labeled examples. To compare, we draw 10 tasks of 32 examples at random from 1% of training data of each downstream dataset; we fine-tune the prompt vectors, that

METHOD	SUB	#TRAINING			MSRVTT-QA	MSVD-QA	ANET-QA	TGIF-QA
		IMG	VID	VQA				
CLIP* [44]		400M	-		2.1	7.2	1.2	3.6
RESERVE [59]	✓	-	20M		5.8	-	-	-
LAVENDER [34]		3M	2.5M		4.5	11.6	-	16.7
Flamingo-3B [1]		2.3B	27M		11.0	27.5	-	-
Flamingo-9B [1]		2.3B	27M		13.7	30.2	-	-
Flamingo [1]		2.3B	27M		17.4	35.6	-	-
FrozenBiLM [57]	✓	-	10M		16.7	33.8	<b>25.9</b>	41.9
Just Ask [55]		69M	-	✓	2.9	7.5	12.2	-
Just Ask [56]		69M	3M	✓	5.6	13.5	12.3	-
BLIP [32]		129M	-	✓	19.2	35.2	-	-
ViTiS (Ours)		-	2.5M		<b>18.2</b>	<b>36.2</b>	25.0	<b>45.5</b>
ViTiS (Ours)	✓	-	2.5M		18.1	36.1	25.5	<b>45.5</b>

Table 7: *Zero-shot VideoQA* top-1 accuracy on test sets, except TGIF-QA on the validation set. Number of pretraining data: image-text/video-text pairs. SUB: subtitle input. VQA: visual question answer pairs. ANET-QA: ActivityNet-QA. CLIP: CLIP ViT-L/14. Flamingo: Flamingo-80B. We gray out methods trained on VQA pairs, which are not directly comparable. \*: CLIP results taken from [57].

METHOD	#SHOT	#PRE-TRAINING			MSRVTT-QA	MSVD-QA	ANET-QA	TGIF-QA
		IMG	VID	#PARAM				
Flamingo-3B [1]	32	2.3B	27M	1.4B	25.6	42.6	-	-
Flamingo-9B [1]	32	2.3B	27M	1.8B	29.4	47.2	-	-
Flamingo-80B [1]	32	2.3B	27M	10B	31.0	52.3	-	-
ViTiS (Ours)	32	-	2.5M	101M	27.0±1.0	41.9±0.8	28.7±1.3	52.2±1.2

Table 8: *Few-shot VideoQA in-context learning*. Mean and standard deviation of top-1 accuracy on test sets, except TGIF-QA on the validation set, over 10 32-shot tasks drawn at random. Only our model involves parameter updates; we fine-tune 0.75M params. Number of pretraining data: image-text/video-text pairs. ANET-QA: ActivityNet-QA.

METHOD	TRAINED MODULES	#TRAINED PARAMS	MSRVTT	MSVD	ANET	TGIF
			-QA	-QA	-QA	-QA
FrozenBiLM [57]	ATP	30M	36.0	46.5	33.2	55.1
ViTiS (Ours)	ATP	101M	36.5	47.6	33.1	55.7
ViTiS (Ours)	Prompts	0.75M	<b>36.9</b>	<b>47.8</b>	<b>34.2</b>	<b>56.2</b>

Table 9: *Few-shot VideoQA* top-1 accuracy on test sets, except TGIF-QA on the validation set. Number of trained parameters: fine-tuned on the downstream dataset, using 1% of training data. ATP: All trainable parameters. ANET-QA: ActivityNet-QA.

is, 0.75M parameters, on each task for 5 epochs and report mean and standard deviation. This can be considered as *test-time prompt tuning* [47] using task-specific annotated data.

Table 8 shows the results of few-shot in-context learning. Flamingo [1] uses a frozen auto-regressive language model with trainable cross-attention layers that incorporate vision and language input, trained on an extreme-scale dataset. The Flamingo-3B, Flamingo-9B, and Flamingo-

80B have 1.4B, 1.8B, and 10B learned parameters, respectively, in addition to the frozen language model. By contrast, our method uses a lighter frozen language model and lighter adaptation modules, resulting in only 101M parameters to learn, and our training data is a relatively small amount of video-text pairs. Despite this, our method outperforms Flamingo-3B [1] on MSRVTT-QA and is on par with MSVD-QA.

## 5. Conclusion

In this work, we explored the adaptation of large-scale pretrained vision and language models for VideoQA under scarcity of data. We introduced multi-modal prompt learning and a visual mapping network to address challenges in such adaptation. Our method consistently outperforms prior works, while requiring minimal parameter fine-tuning in few-shot VideoQA.

**Acknowledgements** This work was granted access to the HPC resources of IDRIS under the allocation 2022-AD011012263R2 made by GENCI.

## References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Proc. NeurIPS*, 2022. 1, 2, 6, 7
- [2] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 2
- [3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proc. ICCV*, 2021. 1, 2, 4, 5
- [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proc. NeurIPS*, 2020. 2
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proc. NeurIPS*, 2020. 6
- [6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *Proc. CVPR*, 2015. 4
- [7] David Chen and William Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proc. ACL*, 2011. 4
- [8] Feng Cheng, Xizi Wang, Jie Lei, David Crandall, Mohit Bansal, and Gedas Bertasius. VindLU: A recipe for effective video-and-language pretraining. In *Proc. CVPR*, 2023. 2
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL*, 2019. 1, 2
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. ICLR*, 2021. 1, 2, 4
- [11] Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. MAGMA—multimodal augmentation of generative models through adapter-based finetuning. In *Proc. Findings of EMNLP*, 2022. 2
- [12] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021. 2
- [13] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. An empirical study of end-to-end video-language transformers with masked visual modeling. In *Proc. CVPR*, 2023. 2
- [14] Noa Garcia, Mayu Otani, Chenhui Chu, and Yuta Nakashima. KnowIT VQA: Answering knowledge-based questions about videos. In *Proc. AAAI*, 2020. 2
- [15] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Proc. CVPR*, 2017. 6
- [16] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. AutoAD: Movie description in context. In *Proc. CVPR*, 2023. 1
- [17] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced bert with disentangled attention. In *Proc. ICLR*, 2021. 1, 2, 4
- [18] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proc. ICML*, 2019. 1, 2, 4
- [19] Jingjia Huang, Yinan Li, Jiashi Feng, Xinglong Wu, Xiaoshuai Sun, and Rongrong Ji. Clover: Towards a unified video-language alignment and fusion model. In *Proc. CVPR*, 2023. 2
- [20] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proc. ICML*, 2021. 1, 2
- [21] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-QA: Toward spatio-temporal reasoning in visual question answering. In *Proc. CVPR*, 2017. 2, 4
- [22] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Proc. ECCV*, 2022. 2
- [23] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *Proc. ECCV*, 2022. 2
- [24] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. MaPLE: Multi-modal prompt learning. In *Proc. CVPR*, 2023. 2
- [25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015. 4
- [26] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proc. EMNLP: System Demonstrations*, 2018. 4
- [27] Yi-Lun Lee, Yi-Hsuan Tsai, Wei-Chen Chiu, and Chen-Yu Lee. Multimodal prompting with missing modalities for visual recognition. In *Proc. CVPR*, 2023. 2
- [28] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proc. CVPR*, 2021. 2
- [29] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. TVQA: Localized, compositional video question answering. In *Proc. EMNLP*, 2018. 2
- [30] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proc. EMNLP*, 2021. 2



- [31] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *Proc. CVPR*, 2022. 2
- [32] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proc. ICML*, 2022. 2, 6, 7
- [33] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: Hierarchical encoder for video+ language omni-representation pre-training. In *Proc. EMNLP*, 2020. 2
- [34] Linjie Li, Zhe Gan, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Ce Liu, and Lijuan Wang. Lavender: Unifying video-language understanding as masked language modeling. In *Proc. CVPR*, 2023. 2, 6, 7
- [35] Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing continuous prompts for generation. In *Proc. ACL*, 2021. 1, 2
- [36] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks". In *Proc. ACL*, 2022. 1, 2, 3
- [37] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. GPT understands, too. *arXiv preprint arXiv:2103.10385*, 2021. 1, 2
- [38] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 1, 2
- [39] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proc. CVPR*, 2022. 1, 2
- [40] Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Lambert Mathias, Marzieh Saeidi, Veselin Stoyanov, and Majid Yazdani. PERFECT: prompt-free and efficient few-shot learning with language models. In *Proc. ACL*, 2022. 2
- [41] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *Proc. ICCV*, 2019. 1, 2
- [42] Ron Mokady, Amir Hertz, and Amit H Bermano. Clip-Cap: CLIP prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 1
- [43] Guanghui Qin and Jason Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proc. NAACL*, 2021. 2
- [44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proc. ICML*, 2021. 1, 2, 4, 7
- [45] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *Technical Report*, 2019. 1, 2
- [46] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. In *Proc. CVPR*, 2023. 2
- [47] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *Proc. NeurIPS*, 2022. 7
- [48] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. In *Proc. CVPR*, 2019. 2
- [49] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. VI-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proc. CVPR*, 2022. 2
- [50] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelwagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding stories in movies through question-answering. In *Proc. CVPR*, 2016. 2
- [51] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Ge Yuying, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. In *Proc. CVPR*, 2023. 2
- [52] Syed Talal Wasim, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Vita-CLIP: Video and text adaptive clip via multimodal prompting. In *Proc. CVPR*, 2023. 2
- [53] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proc. ACM Multimedia*, 2017. 2, 4
- [54] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *Proc. CVPR*, 2016. 4
- [55] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *Proc. ICCV*, 2021. 2, 6, 7
- [56] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Learning to answer visual questions from web videos. *IEEE TPAMI*, 2022. 2, 6, 7
- [57] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. In *Proc. NeurIPS*, 2022. 1, 2, 3, 4, 5, 6, 7
- [58] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-QA: A dataset for understanding complex web videos via question answering. In *Proc. AAAI*, 2019. 2, 4, 6
- [59] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. MERLOT Reserve: Neural script knowledge through vision and language and sound. In *Proc. CVPR*, 2022. 2, 6, 7
- [60] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. MERLOT: Multimodal neural script knowledge models. In *Proc. NeurIPS*, 2021. 1, 2, 4

- [61] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proc. CVPR, 2022*. 2
- [62] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV, 2022*. 2
- [63] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In *Proc. CVPR, 2022*. 2