Prompt Learns Prompt: Exploring Knowledge-Aware Generative Prompt Collaboration for Video Captioning

Liqi Yan[1*] , **Cheng Han**[2*] , **Zenglin Xu**[3] , **Dongfang Liu**[2†] and **Qifan Wang**[4]
[1]Westlake Institute for Advanced Study, Fudan University
[2]Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology
[3]School of Computer Science and Technology, Harbin Institute of Technology
[4]Meta AI, Meta
lqyan18@fudan.edu.cn, {ch7858, dongfang.liu}@rit.edu, wqfcr@fb.com

Abstract

Fine-tuning large vision-language models is a challenging task. Prompt tuning approaches have been introduced to learn fixed textual or visual prompts while freezing the pre-trained model in downstream tasks. Despite the effectiveness of prompt tuning, *what do those learnable prompts learn* remains unexplained. In this work, we explore whether prompts in the fine-tuning can learn knowledge-aware prompts from the pre-training, by designing two sets of prompts — one in pre-training and the other in fine-tuning. Specifically, we present a Video-Language Prompt tuning (**VL-Prompt**) approach for video captioning, which first efficiently pre-train a video-language model to extract key information (*e.g.*, actions and objects) with flexibly generated Knowledge-Aware Prompt (KAP). Then, we design a Video-Language Prompt (VLP) to utilize the knowledge from KAP and fine-tune the model to generate full captions. Experimental results show the superior performance of our approach over several state-of-the-art baselines. We further demonstrate that the video-language prompts are well learned from the knowledge-aware prompts.

1 Introduction

In 1959, three computer science pioneers envisioned that AI *is to create a computer program that simulated human problem-solving behavior*. Humans can process novel tasks effortlessly by using existing knowledge and learning from new information. Artificial systems, however, need a heavily pre-trained base model (e.g., CNNs [He *et al.*, 2016] and Transformers [Dosovitskiy *et al.*, 2021; Wang *et al.*, 2022a]) with extensive fine-tuning on curated data for each problem. This practice is common in deep learning. But adapting these large-scale models to downstream tasks is hard. Full fine-tuning requires storing and deploying a separate copy of the backbone parameters for every task, which is expensive and impractical, especially for modern Transformer-based architectures such as Swin-L [Liu *et al.*, 2021d] (284M parameters), ViT-Huge [Dosovitskiy *et al.*, 2021] (632M parameters), and iGPT-L [Chen *et al.*, 2020] (1362M parameters). Inspired by human-like learning, a substantial amount of concurrent scientific efforts for intelligent systems have been devoted to the development of a novel training strategy in both natural language processing (NLP) and computer vision (CV) to efficiently transfer knowledge across domains.

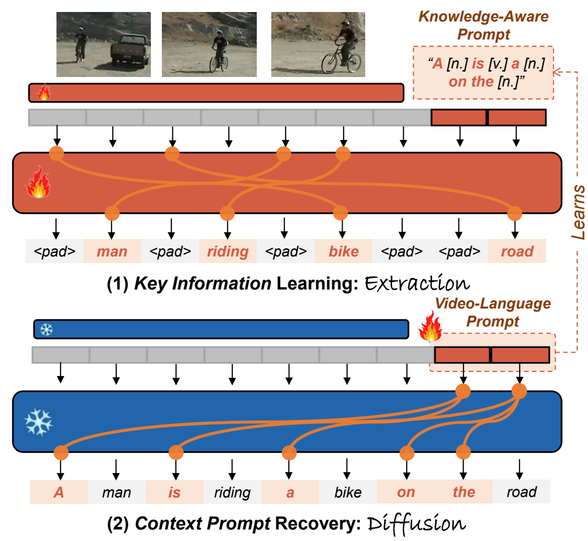

Figure 1: Our method includes two phases: 1) ***Key Information Learning***: model pre-training to extract key information for each video with a knowledge-aware prompt. 2) ***Context Prompt Recovery***: model fine-tuning to recover the pre-trained knowledge-aware prompt for each video and generate the full caption.

This appetite for training has been successfully addressed in natural language processing (NLP) by prompt tuning. The solutions are based on generative language modeling in GPT [Alec and Karthik, 2018] and masked language pre-training in BERT [Devlin *et al.*, 2018]. The idea is to reformulate downstream tasks to look more like those solved during the language model (LM) pre-training with the help of a textual prompt [Liu *et al.*, 2021b]. For example, when recognizing the emotion of a social media post, we may continue with a prompt "*I felt so* __", and ask the LM to fill

*Equal contributions.
†Corresponding author.

the blank with an emotion-bearing word. Creating and experimenting with these prompts takes time and experience, so methods have been proposed to automate the template design process. These methods can be separated into two broad types: a) *discrete prompts*, which automate search for templates described in a discrete space, usually corresponding to natural language phrases [Zhengbao *et al.*, 2019; Shin *et al.*, 2020], and b) *continuous prompts*, which are directly described in the embedding space [Li and Liang, 2021; Lester *et al.*, 2021]. These emerging methods enable quick training of generalizable NLP models containing over one hundred billion parameters for novel tasks [Liu *et al.*, 2021c].

Prompt tuning is a generic form of prefix virtual tokens construction that is natural and applicable in computer vision as well [Lester *et al.*, 2021] [Li and Liang, 2021]. Visual Prompt Tuning (VPT) [Jia *et al.*, 2022] introduces only a small amount of trainable parameters into the image feature input space while keeping the model backbone frozen. Several CLIP-based [Radford *et al.*, 2021] methods adopt VPT-like architectures into their image encoders [Uzair Khattak *et al.*, 2022; Huang *et al.*, 2022], achieving impressive performance in terms of efficiency and accuracy. Researchers recently began to investigate how to jointly optimize prompts across vision and language. For example, UPT [Zang *et al.*, 2022] trains a tiny neural network to generate the prompt for CLIP text and visual encoders, both of them are started with a shared initial prompt. MVLPT [Shen *et al.*, 2022] finds that many target tasks can benefit each other from sharing prompt vectors and thus can be simultaneously learned via multitask prompt tuning.

Despite significant interest in this idea following the triumph of VPT, however, prompt tuning methods for cross-domain tasks have been lagging and face limitations.: **First**, learnable prompts lack explainability and their embeddings are too abstract to provide a human-understandable explanation. Concurrent work has not explored what these prompts actually learn. **Second**, learnable prompts lack explainability. The concurrent work fails to explore what those learnable prompts learn. The embeddings of these learnable prompts are so abstract that it is difficult to provide a human-understandable explanation. **Third**, vision-language Transformer models require extensive self-attention computation, leading to inefficiencies and lack of knowledge transferability due to the heavy parametric architecture. In light of this view, we ask: *how to learn explainable prompts to enable effective learning for across-domain language-vision tasks*?

To address these limitations, we present **VL-Prompt**, a powerful, explainable, and efficient prompt tuning approach. Our contributions are three-fold, as shown in Fig. 1:

- We introduce VL-Prompt, a novel framework for video captioning that splits the task into two parts: a) the *Key Information*, which is extracted from a pre-training module with flexible textual prompts; b) the *Context Prompt*, which is fine-tuned with the frozen pre-trained model. The design enables VL-Prompt to handle the translation of a large-scale vision-language model.
- We propose Knowledge-Aware Prompt (KAP) and Vision-Language Prompt (VLP) to investigate the explainability of the learned prompts. KAP uses syntactic knowledge to guide sentence generation in pre-training. VLP inserts context prompts between keywords to decode captions in fine-tuning. VLP recovers KAP's information and learns the mutual information in the key information.
- Our method has two main benefits. First, in pre-training, it trains the Transformer model to extract key information with sparse attentions, simplifying the model. Second, in fine-tuning, it can handle more frames on a limited GPU memory, because the main network is frozen and does not need gradient storage.

VL-Prompt is an intuitive yet general video-language framework; it is compatible with different video-language network architectures and tasks. We experimentally show: In **§4.2**, with efficient video-language Transformer [Tay *et al.*, 2020], VL-Prompt outperforms other Transformer-based counterparts, O2NA [Liu *et al.*, 2021a] and SwinBERT [Lin *et al.*, 2022], $\uparrow$ **7.5** $\sim$ **31.7** in terms of CIDEr and $\uparrow$ **5.3** $\sim$ **8.1** in terms of B@4, on MSVD benchmark. In **§4.4**, VL-Prompt acquires further improvement by applying our pre-training with KAP on a larger dataset and tuning into a smaller dataset, *i.e.*, for MSVD benchmark, pre-trained on MSR-VTT ($\uparrow$ **0.6** in B@4, $\uparrow$ **0.2** in M), and pre-trained on VATEX ($\uparrow$ **1.4** in B@4, $\uparrow$ **0.4** in M). These results are particularly impressive, considering the number of parameters in the tunable prompt is very small. We hope this work could bring fundamental insights into related fields.

2 Related Work

2.1 Video Transformer and Video Captioning

Recently, base on the impressive performance of Vision Transformers (ViT) [Dosovitskiy *et al.*, 2021], TimeSformer [Bertasius *et al.*, 2021] and ViViT [Arnab *et al.*, 2021] are two popular video Transformer method. More recently, after Swin Transformer [Liu *et al.*, 2021d] is introduced as a general-purpose vision backbone for image understanding, Video Swin Transformer [Liu *et al.*, 2022] extends the scope of local attention computation from only the spatial domain to the spatio-temporal domain. Traditional video captioning methods are based on CNN-RNN structure, including S2VT [Venugopalan *et al.*, 2015], PickNet [Chen *et al.*, 2018], OA-BTG [Zhang and Peng, 2019], SAAT [Zheng *et al.*, 2020], ORG-TRL [Zhang *et al.*, 2020], GLR [Yan *et al.*, 2022a], *etc*. Most recently, SwinBERT [Lin *et al.*, 2022] claims to be the first end-to-end fully Transformer-based model for video captioning.

2.2 Prompt Tuning

Prompt tuning methods can be divided into two categories.

Discrete Prompt Tuning can tune themselves for different tasks by manually constructing different prompts (e.g., CLIP [Radford *et al.*, 2021]) like "*A photo of a [object]*". This strategy has been widely used in recent researches, for example, ALPRO [Li *et al.*, 2021] and STALE [Nag *et al.*, 2022] proposes a prompt of "*A video of [ENTITY]*" for video classification and action detection. A good design for prompt

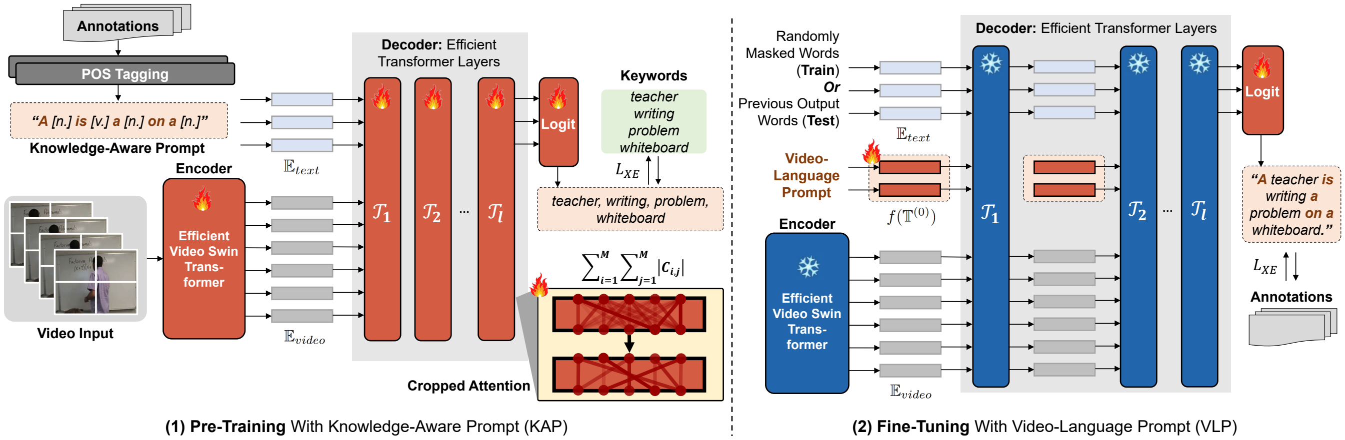

Figure 2: **The framework of our proposed method.** The knowledge-aware prompt is a discrete prompt, while the video-language prompt is a continuous prompt. We use the continuous prompt to learn the discrete prompt.

sentence can achieve a better performance. For example, DetPro [Du *et al.*, 2022] judge whether a predicted object detection box is good by two types of prompt: given a ground truth bounding box of an object class, it says "*a photo of [CLASS]*"; while given a foreground proposal of a partial object, it would instead say "*a photo of partial [CLASS]*". Other methods such as KDDAug [Chen *et al.*, 2022] design different strategies to generate prompts for different VQA question types (*e.g.*, "*[NUMBER] [OBJECT] are there*" when asking the number of an object). All of these methods generate fixed prompts with invariant structure.

Continuous Prompt Tuning automates the process by learning soft prompts (e.g., embeddings). For example, CoOp [Zhou *et al.*, 2022b] tries to learn a prompt content optimization for CLIP text encoder on image classification tasks. CoCoOp [Zhou *et al.*, 2022a] optimizes that learnable text prompt contents by the output of the CLIP image encoder. TPT [Shu *et al.*, 2022] learns textual prompts in a zero-shot manner, via different augmented views of a single test image. For video tasks, [Ju *et al.*, 2022] attempts to learn prompt vectors for CLIP text encoder on some simple video understanding tasks including action classification and localisation. Recently, VPT [Jia *et al.*, 2022] introduces only a small amount of trainable visual prompts in the input space while keeping the model backbone frozen. MaPLe [Uzair Khattak *et al.*, 2022] uses VPT to fine-tune the text encoder and image encoder in the CLIP [Radford *et al.*, 2021]) model. VoP [Huang *et al.*, 2022] also inserts prompt embeddings into the CLIP network for text-video retrieval. However, to the best of our knowledge, all these methods have not explored vision-language generative tasks, and none of them have explored what the prompts have learned.

3 VL-Prompt

3.1 Overview

The task of video captioning is to generate a text sequence that summarizes a given video. In this work, we propose a video-language prompt tuning approach for effective caption generation. The overall model architecture of VL-Prompt is shown in Fig. 2, which consists of two key phases, pre-training with Knowledge-Aware Prompt (KAP) and fine-tuning with Video-Language Prompt (VLP). Intuitively, in the pre-training phase, the video encoder and decoder effectively learns the complex knowledge (the "difficult" part) about the actions and objects from the videos, with the guidance of textual knowledge-aware prompts generated from the annotations. During fine-tuning, the model only needs to tune a few trainable video-language prompts representing the structure of the caption (the "easy" part) while freezing the well-learned encoder and decoder. We adopt an Video Swin Efficient Transformer [Lin *et al.*, 2022] as the video encoder, while the decoder is constructed by a stack of Efficient Transformer Layers [Tay *et al.*, 2020].

3.2 Pre-Training with Knowledge-Aware Prompt

The main responsibilities of the pre-training module is to learn an efficient and effective video encoder for video representation. To this end, we introduce a novel pre-training approach with knowledge-aware prompts.

Textual Knowledge-Aware Prompt. A textual KAP is a discrete prompt which is automatically generated from the annotations of caption. For example, the KAP from original caption of "*A teacher is writing a mathematical problem on a whiteboard in a classroom*" is "*A __ is writing a mathematical __ on a __ in a __*". The pipeline consists of the following two steps: 1) Predict Part-Of-Speech (POS) tags on those annotations via POS tagging model (*e.g.*, FLAIR [Akbik *et al.*, 2019]); 2) Select important types of POS tags (*e.g.*, noun or verb) and mask the corresponding words in the original caption; 3) The knowledge-aware prompt $X = \{x_1, x_2, ..., x_m\}$ (*a.k.a.*, function words) contains remaining words in the caption. The purpose of pre-training is to learn effective video encoder and decoder by recovering the sequence of the marked out keywords $Y = \{y_1, y_2, ..., y_n\}$ with the guidance of the KAP. Essentially, KAP helps reduce the search space of the target words, which provides additional supervision for video captioning.

Sparse Attention Learning. Dense attention for video features in the Transformer decoder is very computationally intensive. To remove unnecessary attention and refine efficient features in the model, the decoder is supervised by the above keywords to learn the sparse attention patterns [Beltagy *et al.*, 2020; Zaheer *et al.*, 2020; Qin *et al.*, 2023; Wang *et al.*, 2022b], which reduces redundancy among the learned video representation. Following SwinBERT [Lin *et al.*, 2022], the attention mask is defined as a learnable matrix with a size of $M \times M$, where M is the length of the video embedding from the encoder. Each value $C_{i,j}$ in this matrix indicates the attention connection between the i_{th} position of the input video embedding and the j_{th} position of the output. This matrix is trained to be more sparse, with the loss designed to reduce the percentage of non-zero elements in the attention connection maps.

Learning Objective. To sum up, the final training objective of the pre-training model is defined as the Cross-Entropy (XE) loss collaborated with the number of non-zero elements in the attention connection maps:

$$arg\min\left[\sum_{y\in\mathcal{Y}} p(\hat{y})\log p(y|X) + \omega \cdot \sum_{i=1}^{M}\sum_{j=1}^{M}|C_{i,j}|\right] \quad (1)$$

where y and $\hat{y}$ denotes the predicted words and the ground-truth respectively, and $\mathcal{Y}$ represents the set of notional words (*e.g.*, noun or verb) that removed from the sentence in the KAP. The weight of attention connection count ω will be analyzed in the experiments.

3.3 Fine-Tuning with Video-Language Prompt

With the limited memory of the GPU, it is often impossible to sample a large number of frames from the video when training a large video-language model. To reduce GPU memory usage and increase the model capability for processing more frames, inspired by VPT [Jia *et al.*, 2022], we propose to fine-tune the pre-trained large-scale model with Video-Language Prompt (VLP), which introduces only a small amount of trainable parameters while keeping the model backbone frozen. We transfer the VPT into a multi-modal prompt tuning framework via designing video-language prompt tokens into the collection of text embeddings and video embeddings. We denote the collection of text embeddings and video embeddings as $\mathbb{E}_{text}$ and $\mathbb{E}_{video}$ respectively.

Video-Language Prompt. The prompt content is defined as a learnable d-dimensional vector as $T_j \in \mathbb{R}^d$ for the j_{th} prompt token. Those prompt tokens are fed into a fully connected linear layer $f(\cdot)$ to get video-language prompt embeddings. Following VPT, our tuning process with VLP also has two variants, VLP-Shallow and VLP-Deep.

VLP-Shallow. Prompts are inserted into the first Transformer layer $\mathcal{T}_1$ only. Each video-language prompt is a learnable d-dimensional vector. A collection of N prompt tokens is denoted as $\mathbb{T}^{(0)} = \{T_j = (t_0, t_1, ..., t_i, ..., t_d) \mid t_i \in \mathbb{R}\}_{j=1}^{N}$, the shallow-prompted decoder with l Transformer layers is defined as:

$$\begin{aligned}
[\mathbb{E}_{text}^{(1)}, \mathbb{E}_{video}^{(1)}, \mathbb{E}_{prompt}^{(1)}] &= \mathcal{T}_1([\mathbb{E}_{text}^{(0)}, \mathbb{E}_{video}^{(0)}, f(\mathbb{T}^{(0)})]) \\
[\mathbb{E}_{text}^{(k)}, \mathbb{E}_{video}^{(k)}, \mathbb{E}_{prompt}^{(k)}] &= \mathcal{T}_k([\mathbb{E}_{text}^{(k-1)}, \mathbb{E}_{video}^{(k-1)}, \mathbb{E}_{prompt}^{(k-1)}]) \\
Z &= Logit(\mathbb{E}_{text}^{(l)})
\end{aligned} \quad (2)$$

where $\mathbb{E}_{prompt}^{(k)}$ represents the prompt embeddings computed from the k_{th} Transformer layer ($1 \leq k < l$), and $Logit$ indicates the Transformer head which convert embeddings into the log-likelihood scores Z for word prediction.

VLP-Deep. Prompts are introduced at *every* Transformer layer's input space. For k_{th} Transformer Layer $\mathcal{T}_k$, we denote the collection of input learnable prompts as $\mathbb{T}^{(k)} = \{T_j = (t_0, t_1, ..., t_i, ..., t_d) \mid p_i \in \mathbb{R}\}_{j=1}^{N}$. As shown in Fig. 2, the deep-prompted decoder with l Transformer layers is formulated as:

$$\begin{aligned}
[\mathbb{E}_{text}^{(k)}, \mathbb{E}_{video}^{(k)}, \mathbb{E}_{prompt}^{(k)}] &= \mathcal{T}_k([\mathbb{E}_{text}^{(k-1)}, \mathbb{E}_{video}^{(k-1)}, f(\mathbb{T}^{(k-1)})]) \\
Z &= Logit(\mathbb{E}_{text}^{(l)})
\end{aligned} \quad (3)$$

Different colors indicate learnable and frozen parameters.

Fine-tuning Objective. During fine-tuning, the sparse attention mask is also frozen. The VLP linear projection and the decoder head are trainable. Following BERT [Devlin *et al.*, 2018], the ground-truth text sentence $\hat{Z}$ is randomly masked (*e.g.*, "*[MASK] teacher is [MASK] a [MASK] problem on [MASK] whiteboard in [MASK] classroom*") and the model prediction Z try to store the masked tokens. Therefore, the fine-tuning objective with VLP becomes:

$$arg\min\sum_{z\in\mathcal{Z}} p(\hat{z})\log p(z) \quad (4)$$

where z and $\hat{z}$ denotes predicted words and the ground-truth respectively. $\mathcal{Z}$ denotes all possible words in vocabulary.

3.4 Theoretical Analysis - Prompt Learns Prompt

The core idea of our system is that different words carry different amounts of information. For example, a frequent word (*e.g.*, function words like "on") carries very little information, while a rare word (*e.g.*, notional words like "teacher") is much more informative. In the following analysis, we show that the entropy of the video-language prompt (VLP) and knowledge-aware prompt (KAP) are equivalent, indicating that our model is able to transfer the prompt knowledge from KAP to VLP.

Assuming that the KAP only contains function words with all notional words (nouns and verbs) removed. Mathematically, given an input video V and a knowledge-aware prompt X, our pre-training model M is trained to output a sequence of notional words Y, as shown in Fig. 3. Those function and notional word sequences can be represented by these two random variables. According to the Shannon's theory [Shannon, 1948], the information content or entropy of the pre-trained model can be defined as a conditional entropy:

$$\begin{aligned}
H(M) &= H(Y|X, V) \\
&= -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y, V)\log\frac{p_{X,Y}(x, y, V)}{p_X(x)\cdot p(V)}
\end{aligned} \quad (5)$$

where $\mathcal{X}$ and $\mathcal{Y}$ denote the set of function words and notional words respectively. When fine-tuning the model with the video-language prompt network F, which consists of the learnable vectors $\mathbb{T}$ and the linear layer $f(\cdot)$, the final sentence output Z can be viewed as a combination of X and Y (*i.e.*, Z is the union distribution of X and Y). We denote

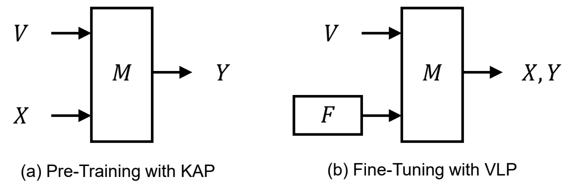

Figure 3: Theoretical analysis of our pre-training and fine-tuning models. V denotes the input video. M denotes the pre-training model including the encoder and the decoder. The prompt network F contains the learnable prompt tokens $\mathbb{T}$, and the linear layer $f(\cdot)$. X and Y indicates the knowledge-aware prompt which consists of function words and the predicted notional words respectively.

$p_Z(z) = p_{X,Y}(x, y)$. Then the joint entropy of the full model is given by:

$$\begin{aligned} H(M, F) &= H(X, Y|V) \\ &= -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y, V) \log \frac{p_{X,Y}(x, y, V)}{p(V)} \end{aligned} \quad (6)$$

Then we can measure the expected entropy of the video-language prompt F in condition of M as:

$$\begin{aligned} H(F) &= H(M, F) - H(M) + I(M; F) \\ &= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y, V) \log \frac{p_{X,Y}(x, y, V)}{p(V)} \\ &- \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y, V) \log \frac{p_{X,Y}(x, y, V)}{p_X(x) \cdot p(V)} + I(M; F) \\ &= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y, V) \log \frac{1}{p_X(x)} + I(M; F) \end{aligned} \quad (7)$$

where $I(M; F)$ denotes the mutual information between the distribution of the model M and the prompt parameters F. Since M is pre-trained to predict notional words with the guidance of function words and then frozen in the fine-tuning phase, the mutual information $I(M; F)$ is very small. Therefore, we can approximate the expected information of our proposed video-language prompt F to the information of knowledge-aware prompt X, according to the information theory [Thomas and Joy, 2006]:

$$\begin{aligned} H(F) &\approx \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y, V) \log \frac{1}{p_X(x)} \\ &= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y, V) \log \frac{p_{X,Y}(x, y, V)}{p_X(x)} \\ &- \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y, V) \log p_{X,Y}(x, y, V) \\ &= -H(y, V|x) + H(x, y, V) = H(X) \end{aligned} \quad (8)$$

From the above analysis, we show that the information contents in video-language prompt (VLP) and knowledge-aware prompt (KAP) are equivalent, indicating that VLP learns from KAP, to guide the recovery of the full caption. The full sentence recovery can be viewed as a diffusion process, where function words are diffused among those predicted notional words and then restored.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on two public video captioning datasets: a) **MSR-VTT** [Xu *et al.*, 2016] consists of 10K video clips. Each video clip has 20 ground-truth captions. We use the standard split, which has 6.5K training videos, 497 validation videos and 2.9K testing videos. b) **MSVD** is a collection of 2K video clips downloaded from YouTube. Each video clip has roughly 40 ground-truth captions written by humans. Similar to the prior articles [Chen *et al.*, 2018; Zheng *et al.*, 2020], we use the standard split which contains 1.2K training videos, 100 validation videos, and 670 test videos. We further leverage a larger dataset, **VATEX**, to study the effect of pre-training. **VATEX** contains 41.3K videos. Each video clip has 20 ground-truth captions. We use the official training set for training, and evaluate the results using the public test set.

Implementation Details. The number of the Swin Transformer layers in the video encoder is set to 3. The output length of the video feature embedding from the encoder is set to $M = 392$. The dimension of the video feature embedding from the encoder is 768, while the dimension of the hidden state of the decoder is 512. To ensure the same dimensionality of the video embedding in the encoder and decoder, we transform the video embedding using a linear fully connected network. The number of the transformer layers in the decoder is set to 11. The dimension of the video-language prompt is set to $d = 1024$. The number of the video-language prompt tokens is set to 100. The max length of the text sequence is set to 50. The max epoch number is set to 10 for both pre-training and fine-tuning. The learning rate is set to 10^{-4} in pre-training, and 10^{-5} in fine-tuning.

4.2 Main Results

We compare our **VL-Prompt** with several state-of-the-art methods on the above commonly used benchmarks. Following previous research [Yan *et al.*, 2022b; Lin *et al.*, 2022], we provide detailed comparisons using a diverse set of performance metrics, including BLEU4 [Papineni *et al.*, 2002], METEOR [Banerjee and Lavie, 2005], ROUGE-L [Lin and Och, 2004] and CIDEr [Vedantam *et al.*, 2015]. Table 1 shows detailed comparisons with eleven LSTM-based and five transformer-based methods for video captioning. In our method, we sample a large number of frames ($F = 64$) from each video, since our Efficient VL Transformer can avoid overly dense features. It can be seen from the results that: a) Compared to those methods using 2D and 3D CNN encoders, Transformer based method can jointly learn the features of 2D appearance and 3D motion. For example, the popular SAAT [Zheng *et al.*, 2020] uses spatial and temporal feature to represent the static scene and the dynamic motions, but it fails to jointly train the 2D and 3D representation, which can be learned via a visual Transformer. Thus, on MSR-VTT `test`, our method outperforms SAAT ↑ **4.3** in B@4, ↑ **2.4** in M, ↑ **1.5** in R, and ↑ **4.3** in terms of CIDEr. b) Compared to those methods using Transformer as decoder, prompt tuning methods rely on the well pre-trained large-scale models, which is difficult to be trained. SwinBERT [Lin *et al.*, 2022]

Method	Encoder		Decoder	PT	F	MSR-VTT				MSVD			
	2D Appearance	3D Motion				B@4	M	R	C	B@4	M	R	C
SA-LSTM [Xu *et al.*, 2016]	VGG	C3D	LSTM	×	16	36.3	25.5	58.3	39.9	45.3	31.9	64.2	76.2
RecNet [Wang *et al.*, 2018a]	VGG	C3D		×	-	-	26.6	59.3	42.7	52.3	34.1	69.8	80.3
ORG-TRL [Zhang *et al.*, 2020]	IncepResnetV2	C3D		×	28	43.6	29.7	62.1	50.9	54.3	36.4	73.9	95.2
STGraph [Pan *et al.*, 2020]	ResNet-101	I3D		×	10	40.5	28.3	60.9	47.1	52.2	36.9	73.9	93.0
SGN [Ryu *et al.*, 2021]	ResNet-101	3D-ResNext		×	30	40.8	28.3	60.8	49.5	52.8	35.5	72.9	94.3
RCG [Zhang *et al.*, 2021]	ResNet-152	I3D		×	28	42.8	29.3	61.7	52.9	-	-	-	-
HRL [Wang *et al.*, 2018b]	ResNet-152	-		×	-	41.3	28.7	61.7	48.0	-	-	-	-
PickNet [Chen *et al.*, 2018]	ResNet-152	-		×	7	38.9	27.2	59.5	42.1	46.1	33.1	69.2	76.0
POS$_{RL}$ [Wang *et al.*, 2019]	IncepResnetV2	I3D		×	64	41.3	28.7	62.1	53.4	53.9	34.9	72.1	91.0
SAAT [Zheng *et al.*, 2020]	IncepResnetV2	C3D		×	28	39.9	27.7	61.2	51.0	46.5	33.5	69.4	81.0
HMN [Ye *et al.*, 2022]	IncepResnetV2	C3D		×	16	43.5	29.0	62.7	51.5	59.2	37.7	75.1	104.0
GL-RG [Yan *et al.*, 2022b]	ResNeXt-101	3D-Resnet-18		×	30	42.9	29.9	62.2	54.3	60.5	38.9	76.4	101.0
O2NA [Liu *et al.*, 2021a]	ResNet-101	3D-ResNext	Trans.	×	8	41.6	28.5	62.4	51.1	55.4	37.4	74.5	96.4
SwinBERT [Lin *et al.*, 2022]	Video Swin Transformer			×	64	41.9	29.9	62.1	53.8	58.2	41.3	77.5	120.6
VPT [Jia *et al.*, 2022]	ViT			✓	32	41.2	27.9	61.5	50.3	54.6	36.0	73.1	94.7
Prompting [Ju *et al.*, 2022]	CLIP Image Encoder			✓	16	42.0	28.8	62.3	52.2	56.4	38.6	75.4	99.8
VoP [Huang *et al.*, 2022]	CLIP Image Encoder			✓	12	42.1	28.9	61.9	51.7	57.9	40.2	76.3	105.9
Ours (VL-Prompt)	Efficient VL Transformer			✓	64	**43.2**	**30.1**	**62.7**	**55.3**	**63.5**	**41.6**	**78.9**	**128.1**

Table 1: **Comparisons with state-of-the-art methods** on MSR-VTT `test` and MSVD `test`. PT means prompt tuning. F denotes the number pf sampled frames per video. All of those VPT-based image or video analysis methods are transferred into the video captioning task via replacing the [cls] token into the text sequence token. Our method is first pre-trained on 16 frames and then tuned on 64 frames.

Method	B@4	M	R	C	training fps
VL-Prompt	63.5	41.6	78.9	128.1	7.7
w/o KAP	53.1	34.6	72.4	102.9	8.1
w/o VLP	59.3	37.8	76.1	115.6	8.0

Table 2: **Impact of KAP and VLP** on MSVD `test`.

introduces a BERT-based training method with sparse attention, which indeed benefits the pre-training of the large-scale video-language models, but fails to explore the inner relationships over the semantic context. Thus, on MSVD `test`, our method outperforms SwinBERT ↑ **5.3** in B@4, ↑ **0.3** in M, ↑ **1.4** in R, and ↑ **7.5** in terms of CIDEr. c) VL-Prompt outperforms those prompt tuning baselines, indicating the effectiveness of our prompt design and learning. The learned video-language prompts are more explainable [Wang *et al.*, 2023], which effectively decouple the structured knowledge in the captions and restore them efficiently.

4.3 Ablation Study

We conduct a comprehensive ablation study on MSVD and MSR-VTT benchmarks to investigate the capability of the proposed model.

Impact of KAP and VLP. To analyze the impact of KAP and VLP, we conduct ablation experiments on two variants by removing KAP and VLP from the VL-Prompt respectively. As reported in Table 2, we observe significant performance drop after removing either type of prompts, indicating the importance of both KAP and VLP in the model. This demonstrates that our VLP prompts can capture the internal relations, which denote the semantic correlation between tokens within a sentence, by applying reasoning and linking the input keywords (nouns and verbs) together. Nevertheless, our VL-Prompt with both prompts achieves the best performance with slightly lower training fps.

Masking	B@4	M	R	C	Attn.	training fps
KAP	63.5	41.6	78.9	128.1	12%	7.7
Alternate	56.8	36.9	74.8	104.3	59%	7.5
Random	57.7	37.8	75.6	106.2	31%	7.5
Noun	58.7	41.0	77.8	118.2	10%	7.8
Verb	58.8	41.2	77.7	117.7	8%	7.9

Table 3: **Different masking strategies** of KAP on MSVD `test`. "Alternate" denotes masking words alternately (*e.g.*, $1_{st}, 3_{rd}, 5_{th}$, *etc.*). "Random" strategy is to randomly mask words. "Noun" and "Verb" denote only masking nouns and verbs, respectively.

Method	B@4	M	R	C	fps
Full Fine-Tuning	65.2	41.3	80.6	147.0	7.6
Linear Probing	62.3	41.1	80.1	140.3	8.0
VLP-Shallow	65.5	42.0	80.3	150.2	8.1
VLP-Deep	66.4	42.7	81.3	153.8	7.9

Table 4: **Different fine-tuning strategy** on MSVD `val`. The pre-training with KAP and a frame number of 64 is used in all methods.

Prompt Design in KAP. We further conduct experiments of different masking strategies in learning context prompt, including alternative masking, random masking, noun masking and verb masking, to understand the effectiveness of different prompt designs. Table 3 shows the ablation results on MSVD. It can be seen that our original KAP (Noun and Verb masking) outperforms all other strategies with big margins, while only masking nouns or verbs achieves a faster speed. This is consistent with our expectation as these nouns and verbs are the most informative words in the caption, representing the objects and actions in the video.

VLP-Shallow vs. VLP-Deep vs. Full. We compare the results of different VLPs with Linear Probing and Full fine-tuning in Table 4. It is clear that VLP-Deep outperforms all the other parameter-efficient tuning protocols, including VLP-Shallow, Linear Probing, and even outperforms full

D \ L	10	20	50	100	200
256	139.9	141.6	143.1	145.4	148.5
512	142.7	143.5	148.0	152.9	153.7
1024	146.3	149.8	152.1	153.8	153.4
2048	147.2	150.1	152.9	153.0	153.6

Table 5: **The ablation study of VLP** on MSVD `val`. 'D' denotes the dimension of VLP and 'L' means the length of VLP.

Method	F in Pre-Train	F in Fine-Tune				
		4	8	16	32	64
w/o KAP	4	125.3	126.8	127.6	129.1	131.1
	16	131.5	134.1	136.0	138.2	139.7
VL-Prompt	4	136.0	139.9	146.5	150.3	152.7
	16	135.8	137.2	144.1	149.6	153.8

Table 6: **The comparison of different number of frames** F on MSVD `val`. CIDEr is used as the evaluation metric.

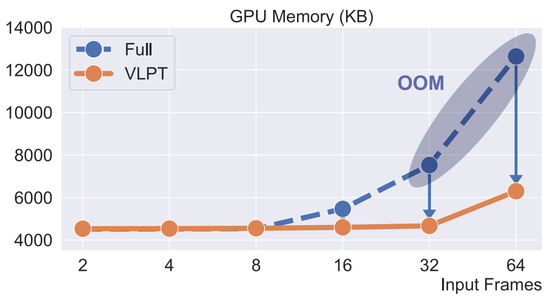

Figure 4: **The memory used per GPU when the model is fine-tuned with different strategy.** The batch size is fixed to 1. When full fine-tuning is adopted, the $F = \{32, 64\}$ frames will cause an Out-Of-Memory (OOM) error with a large batch size. In contrast, the VLP can reduce the GPU memory usage since the gradients of the Transformer parameters do not need to be saved.

fine-tuning. VLP-Shallow achieves the fast speed with a comparable performance over all methods. Similar observations have been found in VPT [Jia *et al.*, 2022].

Prompt Length & Dimension. To investigate the impact of prompt length and dimension in VLP, we conduct ablation study of different combinations of VLP length and dimension. As shown in Table 5, the best choice for prompt tokens is a length of 100 and a dimension of 1024.

Sampled Frame Number. We uniformly sample $F = \{2, 4, 8, 16, 32, 64\}$ frames from the given video clip to train and test our method on both MSVD and MSR-VTT datasets. As we increase the number of frames, we observe consistent performance improvements in terms of CIDEr (see Table 6). We also find that without VLP fine-tuning, the network is too large for $F = \{32, 64\}$ which causes an Out-Of-Memory error, as shown in Fig. 4. In contrast, the network works well on that 11GB GPU during fine-tuning with our VLP, demonstrating the efficiency of our VL-Prompt.

4.4 Effect of Pre-training

To further illustrate the effect of our proposed pre-training method, Table 7 shows the performance of VL-Prompt with additional pre-training data. It can be seen that pre-training on a large dataset achieves better performance in all cases, especially when combining with large VATEX dataset in pre-training. For example, VL-Prompt improves the performance by ↑ **1.2** and ↑ **1.7** in terms of B@4 on MSVD through adding MSR-VTT and VATEX to pre-training, respectively. Moreover, we evaluate the efficiency of our approach by comparing it with the baselines. For example, on the MSVD dataset, our model attains an average fps of 7.7, and outperforms SwinBERT and GL-RG in terms of efficiency (which achieve 7.2 and 6.3 fps respectively). This observation validates the effectiveness of pre-training, which is crucial in the later VLP fine-tuning.

Fine-Tune	Add. Data	B@4	M	R	C
MSR-VTT	MSVD	43.6$\uparrow_{0.4}$	30.4$\uparrow_{0.3}$	62.9$\uparrow_{0.2}$	55.4$\uparrow_{0.1}$
	VATEX	44.2$\uparrow_{1.0}$	30.6$\uparrow_{0.5}$	63.4$\uparrow_{0.7}$	55.7$\uparrow_{0.4}$
MSVD	MSR-VTT	64.7$\uparrow_{1.2}$	42.2$\uparrow_{0.6}$	79.4$\uparrow_{0.5}$	128.4$\uparrow_{0.3}$
	VATEX	65.2$\uparrow_{1.7}$	42.4$\uparrow_{0.8}$	79.8$\uparrow_{0.9}$	128.6$\uparrow_{0.5}$

Table 7: **Effects of pre-training with additional data** for our VL-Prompt on MSR-VTT and MSVD `test`. Results show we could further improve the performance by pre-training on a larger dataset. The up-arrow (↑) number indicates the improvement compared to self pre-training (as shown in Table 1).

ω	0	0.1	0.3	0.5	0.7
CIDEr on MSVD	102.9	115.4	119.5	128.1	124.2

Table 8: **Impact of different values of** ω on MSVD `test`. $\omega = 0$ means keeping the attention connection ratio as 100%.

4.5 Impact of ω

To understand the impact of the hyper-parameter ω, we evaluate the model performance by varying the hyper-parameters ω from $\{0, 0.1, 0.3, 0.5, 0.7\}$. The model performances with different hyper-parameter values are reported in Table 8. It can be seen that $\omega = 0.5$ achieves the best performance, indicating that the model needs to identify a good trade-off between efficiency and effectiveness.

5 Conclusion

In this paper, we propose a novel prompt tuning based video captioning approach, VL-Prompt, by designing two different sets of prompts in pre-training and fine-tuning phases respectively. We first pre-train a video-language model to extract key information with the guidance of knowledge-aware prompts. Then, we design a video-language prompt to transfer the knowledge from the knowledge-aware prompts and fine-tune the model to generate full captions. Experimental results show the superior performance of our approach over several state-of-the-art baselines. Theoretical analysis on how the information from knowledge-aware prompts is transferred to the video-language prompts is also conducted. A potential drawback of our method is the low generalization ability that may result from the limited number of parameters and data samples. In future, we plan to investigate VL-Prompt in zero-shot settings. We also plan to apply VL-Prompt in other downstream tasks such as VQA. We hope this work can inspire future studies in video-language prompt tuning.

References

[Akbik *et al.*, 2019] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL*, pages 54–59, 2019.

[Alec and Karthik, 2018] Radford Alec and Narasimhan Karthik. Improving language understanding by generative pre-training. *OpenAI*, 2018.

[Arnab *et al.*, 2021] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021.

[Banerjee and Lavie, 2005] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop*, pages 65–72, 2005.

[Beltagy *et al.*, 2020] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[Bertasius *et al.*, 2021] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[Chen *et al.*, 2018] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. In *ECCV*, 2018.

[Chen *et al.*, 2020] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

[Chen *et al.*, 2022] Long Chen, Yuhang Zheng, and Jun Xiao. Rethinking data augmentation for robust visual question answering. In *ECCV*, pages 95–112, 2022.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

[Du *et al.*, 2022] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, pages 14084–14093, 2022.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Huang *et al.*, 2022] Siteng Huang, Biao Gong, Yulin Pan, Jianwen Jiang, Yiliang Lv, Yuyuan Li, and Donglin Wang. Vop: Text-video co-operative prompt tuning for cross-modal retrieval. *arXiv preprint arXiv:2211.12764*, 2022.

[Jia *et al.*, 2022] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, pages 709–727, 2022.

[Ju *et al.*, 2022] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, 2022.

[Lester *et al.*, 2021] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, pages 3045–3059, 2021.

[Li and Liang, 2021] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL-IJCNLP*, pages 4582–4597, 2021.

[Li *et al.*, 2021] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven C. H. Hoi. Align and prompt: Video-and-language pre-training with entity prompts. *CVPR*, pages 4943–4953, 2021.

[Lin and Och, 2004] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *ACL*, pages 605–612, 2004.

[Lin *et al.*, 2022] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In *CVPR*, 2022.

[Liu *et al.*, 2021a] Fenglin Liu, Xuancheng Ren, Xian Wu, Bang Yang, Shen Ge, and Xu Sun. O2NA: an object-oriented non-autoregressive approach for controllable video captioning. In *ACL/IJCNLP*, 2021.

[Liu *et al.*, 2021b] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 2021.

[Liu *et al.*, 2021c] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *ArXiv*, abs/2103.10385, 2021.

[Liu *et al.*, 2021d] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.

[Liu *et al.*, 2022] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, pages 3202–3211, 2022.

[Nag *et al.*, 2022] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Zero-shot temporal action detection via vision-language prompting. In *ECCV*, 2022.

[Pan *et al.*, 2020] Boxiao Pan, Haoye Cai, De-An Huang, Kuan-Hui Lee, Adrien Gaidon, Ehsan Adeli, and Juan Carlos Niebles. Spatio-temporal graph for video captioning with knowledge distillation. In *CVPR*, 2020.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[Qin *et al.*, 2023] Zheyun Qin, Xiankai Lu, Xiushan Nie, Dongfang Liu, Yilong Yin, and Wenguan Wang. Coarse-to-fine video instance segmentation with factorized conditional appearance flows. *Journal of Automatica Sinica*, 10(5):1–17, 2023.

[Radford *et al.*, 2021] 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 *ICML*, 2021.

[Ryu *et al.*, 2021] Hobin Ryu, Sunghun Kang, Haeyong Kang, and C. Yoo. Semantic grouping network for video captioning. In *AAAI*, 2021.

[Shannon, 1948] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[Shen *et al.*, 2022] Sheng Shen, Shijia Yang, Tianjun Zhang, Bohan Zhai, Joseph E Gonzalez, Kurt Keutzer, and Trevor Darrell. Multitask vision-language prompt tuning. *arXiv preprint arXiv:2211.11720*, 2022.

[Shin *et al.*, 2020] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *EMNLP*, 2020.

[Shu *et al.*, 2022] 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 *NeurIPS*, 2022.

[Tay *et al.*, 2020] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 2020.

[Thomas and Joy, 2006] MTCAJ Thomas and A Thomas Joy. *Elements of information theory*. Wiley-Interscience, 2006.

[Uzair Khattak *et al.*, 2022] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. *arXiv e-prints*, pages arXiv–2210, 2022.

[Vedantam *et al.*, 2015] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[Venugopalan *et al.*, 2015] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *CVPR*, pages 4534–4542, 2015.

[Wang *et al.*, 2018a] Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. Reconstruction network for video captioning. In *CVPR*, 2018.

[Wang *et al.*, 2018b] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, 2018.

[Wang *et al.*, 2019] Bairui Wang, L. Ma, W. Zhang, Wenhao Jiang, Junling Wang, and W. Liu. Controllable video captioning with pos sequence guidance based on gated fusion network. In *ICCV*, 2019.

[Wang *et al.*, 2022a] Qifan Wang, Yi Fang, Anirudh Ravula, Fuli Feng, Xiaojun Quan, and Dongfang Liu. Webformer: The web-page transformer for structure information extraction. In *ACM Web Conference*, 2022.

[Wang *et al.*, 2022b] Wenguan Wang, James Liang, and Dongfang Liu. Learning equivariant segmentation with instance-unique querying. In *NeurIPS*, 2022.

[Wang *et al.*, 2023] Wenguan Wang, Cheng Han, Tianfei Zhou, and Dongfang Liu. Visual recognition with deep nearest centroids. In *ICLR*, 2023.

[Xu *et al.*, 2016] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[Yan *et al.*, 2022a] Liqi Yan, Siqi Ma, Qifan Wang, Yingjie Chen, Xiangyu Zhang, Andreas Savakis, and Dongfang Liu. Video captioning using global-local representation. *TCSVT*, 32(10):6642–6656, 2022.

[Yan *et al.*, 2022b] Liqi Yan, Qifan Wang, Yiming Cui, Fuli Feng, Xiaojun Quan, Xiangyu Zhang, and Dongfang Liu. Gl-rg: Global-local representation granularity for video captioning. In *IJCAI*, 2022.

[Ye *et al.*, 2022] Hanhua Ye, Guorong Li, Yuankai Qi, Shuhui Wang, Qingming Huang, and Ming-Hsuan Yang. Hierarchical modular network for video captioning. In *CVPR*, pages 17939–17948, 2022.

[Zaheer *et al.*, 2020] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *NeurIPS*, 33:17283–17297, 2020.

[Zang *et al.*, 2022] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022.

[Zhang and Peng, 2019] Junchao Zhang and Yuxin Peng. Object-aware aggregation with bidirectional temporal graph for video captioning. In *CVPR*, 2019.

[Zhang *et al.*, 2020] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *CVPR*, 2020.

[Zhang *et al.*, 2021] Ziqi Zhang, Zhongang Qi, Chunfeng Yuan, Ying Shan, Bing Li, Ying Deng, and Weiming Hu. Open-book video captioning with retrieve-copy-generate network. In *CVPR*, 2021.

[Zheng *et al.*, 2020] Qi Zheng, Chaoyue Wang, and D. Tao. Syntax-aware action targeting for video captioning. In *CVPR*, 2020.

[Zhengbao *et al.*, 2019] Jiang Zhengbao, Xu Frank, F., Araki J., and Neubig Graham. How can we know what language models know? *TACL*, 8:423–438, 2019.

[Zhou *et al.*, 2022a] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022.

[Zhou *et al.*, 2022b] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022.