READ-PVLA: Recurrent Adapter with Partial Video-Language Alignment for Parameter-Efficient Transfer Learning in Low-Resource Video-Language Modeling

Thong Nguyen[1], **Xiaobao Wu**[2], **Xinshuai Dong**[3], **Khoi Le**[4], **Zhiyuan Hu**[1], **Cong-Duy Nguyen**[2], **See-Kiong Ng**[1], **Anh Tuan Luu**[2]

[1] Institute of Data Science (IDS), National University of Singapore, Singapore
[2] Nanyang Technological University (NTU), Singapore;
[3] Carnegie Mellon University, USA
[4] VinAI Research, Vietnam

Abstract

Fully fine-tuning pretrained large-scale transformer models has become a popular paradigm for video-language modeling tasks, such as temporal language grounding and video-language summarization. With a growing number of tasks and limited training data, such full fine-tuning approach leads to costly model storage and unstable training. To overcome these shortcomings, we introduce lightweight adapters to the pre-trained model and only update them at fine-tuning time. However, existing adapters fail to capture intrinsic temporal relations among video frames or textual words. Moreover, they neglect the preservation of critical task-related information that flows from the raw video-language input into the adapter's low-dimensional space. To address these issues, we first propose a novel **RE**current **AD**apter (READ) that employs recurrent computation to enable temporal modeling capability. Second, we propose **P**artial **V**ideo-**L**anguage **A**lignment (PVLA) objective via the use of partial optimal transport to maintain task-related information flowing into our READ modules. We validate our READ-PVLA framework through extensive experiments where READ-PVLA significantly outperforms all existing fine-tuning strategies on multiple low-resource temporal language grounding and video-language summarization benchmarks.

1 Introduction

Video-language modeling is a challenging problem since it involves understanding both video and language modalities. For example, temporal language grounding (TLG) model comprehends video detail and language query to localize semantically related video moments (Figure 1 (left)), or video-language summarization (VLS) model extracts information from both video content and language transcript to write the summary (Figure 1 (right)).

Previous video-language modeling methods (Liu et al. 2022; Lei, Berg, and Bansal 2021; Yu et al. 2021) employ pretrained Transformer models such as Unified Multimodal Transformer (UMT) (Liu et al. 2022) and Vision-Guided BART (VG-BART) (Yu et al. 2021), and fine-tune all the parameters of these models for every single task. This results in substantial storage overhead since each task demands storing a separate model (Zhang et al. 2023). Moreover, because of the difficulty of collecting video-language data (Pan et al. 2022), fully fine-tuning these over-parameterized models in low-resource scenarios, where limited training data is available, leads to instability and sub-optimal performance (Jiang et al. 2022; Huang et al. 2023).

To address these shortcomings, adapters are proposed as a parameter-efficient solution for finetuning video-language pretrained transformers (Jiang et al. 2022; Zhang et al. 2023; Yang et al. 2023; Sung, Cho, and Bansal 2022; Chen et al. 2022). The strategy is to add additional adaptation module to each layer of the pre-trained network and only the adaptation modules are trained during fine-tuning to improve the parameter-performance trade-off. These modules rely on non-linear projections to downproject video-language inputs into low-dimensional space then up-project them back to the original high-dimensional space. However, such projections consider video frames and textual words as separate tokens, thus ignoring the intrinsic temporal dependency among video frames or textual words. Without such dependency information, it is difficult to reason about temporal context in the video to properly ground the language (*e.g.* in Figure 1, determine the *expression* of the *girl after*, not *before*, the *proposal*), or coherently link the entities in the summary (*e.g.* in Figure 1, recap the chronological order of *bolt removing* and *puller threading*). Moreover, because at fine-tuning time only adaptation modules are trained using limited video-language data, little attention is paid to the information flow that starts from the raw video-language inputs till the low-dimensional space of the adaptation modules. This may result in losing essential task-related information and carrying noise into these modules (Tsai et al. 2020; Han, Chen, and Poria 2021).

To resolve the first issue, we propose a novel adapter architecture, **RE**current **AD**apter (READ), for video-language modeling tasks. The key idea is to incorporate the recurrent modeling ability into the adaptation module to capture the temporal dependency of video frames and language entities (Goodfellow, Bengio, and Courville 2016). As such, we formulate READ as a parameter-efficient bottleneck with a sequence of operations including feature dimension reduction, recurrent modeling, and feature dimension recovery. Since the incorporated recurrent computation works in the low dimension (*e.g.* 4-dimensional), our READ module stands as a

Figure 1: Examples of the TLG and VLS problems. TLG model needs to understand the meaning of language entities such as *proposal* or *girl*, and the existence of *expression* in video frames. VLS model is expected to recognize salient information, *e.g. crank bolt*, *bottom bracket* from the language, and *bicycle* from the video.

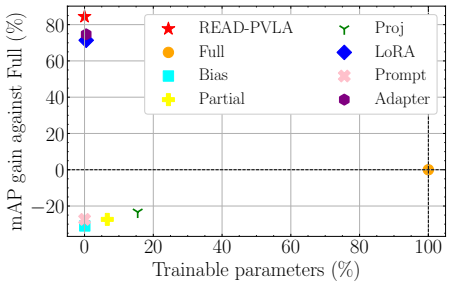

Figure 2: Comparison of our proposed READ method with the full fine-tuning and other parameter-efficient fine-tuning methods. For each method, we denote the mAP gain averaged over the domains of the YouTube Highlights dataset together with the number of trainable parameters.

lightweight design and can be cheaply integrated throughout the Transformer architecture for enhancing video-language modeling, using only up to 1.20% trainable parameters.

As for the second issue, we propose **P**artial **V**ideo-**L**anguage **A**lignment (PVLA), a novel objective to explicitly encourage the alignment between video and language representations, thus capturing invariant aligned information across modalities that are critical for downstream tasks. The key concept is to minimize the Partial Optimal Transport (POT) distance between the distribution over video frame representations and the distribution over textual word representations. The rationale for our partial implementation of optimal transport lies in that video and language do not exhibit complete one-to-one correspondence. Typically, the language does not describe all aspects of the video, and only part of the language sequence is strongly related to part of the video frames, *e.g.* in Figure 1 the language input about the *girl's expression* is only related to the target grounding. As such, utilizing POT for distribution matching is to focus on essential masses that are strongly related between modalities, hence optimizing towards better video-language alignment and gaining more control over video-language information passed into our READ modules.

Based on our novel proposals, we construct the READ-PVLA framework that can be employed to finetune various pre-trained Transformer architectures such as multimodal transformer (UMT (Liu et al. 2022), Moment-DETR (Lei, Berg, and Bansal 2021)), and generative vision-guided transformer models (VG-BART (Lewis et al. 2020) and VG-T5 (Raffel et al. 2020)). Through freezing these pre-trained models and fine-tuning only our READ modules with PVLA objective, we outperform standard fine-tuning and other parameter-efficient methods with substantially fewer tunable parameters (Figure 2) for low-resource video-language tasks, including temporal language grounding and video-language summarization. To sum up, our contributions can be summarized as:

- We propose **RE**current **AD**apter (READ), a novel adapter architecture, that better captures temporal information for modeling video-language tasks.
- We propose **P**artial **V**ideo-**L**anguage **A**lignment (PVLA) objective to encourage the alignment between video and language modalities during the adaptation process.
- We validate our READ-PVLA framework by extensive experiments using multiple low-resource temporal language grounding and video-language summarization datasets, where READ-PVLA outperforms all existing fully or parameter-efficient fine-tuning strategies with only up to 1.20% parameters tunable.

2 Related Work

2.1 Parameter-Efficient Transfer Learning

Recent efforts have sought to propose techniques to reduce the cost of fine-tuning these large-scale models. The techniques can be categorized into three directions. The first one, dubbed as adapter, introduces lightweight modules between Transformer layers that work in low-dimensional space (Houlsby et al. 2019; Pan et al. 2022; Chen et al. 2022; Xu et al. 2023). During fine-tuning, only parameters of the adapters are updated and all of the original Transformer are kept frozen. The second approach, called prompt tuning, appends a sequence of prefix continuous tokens to every input and solely tunes these tokens for adapting to the downstream task (Jia et al. 2022; Huang et al. 2023). The third approach approximates the weight update with low-rank matrices (Hu et al. 2021). Only values of the matrices are learned during training to satisfy the parameter-efficiency requirement.

2.2 Video-Language Modeling

Recent video-language modeling tasks, *e.g.* temporal language grounding (TLG) (Liu et al. 2022; Lei, Berg, and Bansal 2021; Nguyen et al. 2023) or video-language summarization (VLS) (Yu et al. 2021; Liu et al. 2023) have been dominated by deep Transformer models. Regarding the TLG task, (Lei, Berg, and Bansal 2021) collect a query-based benchmark and construct a Transformer model pre-trained upon automatic speech recognition to tackle not only their

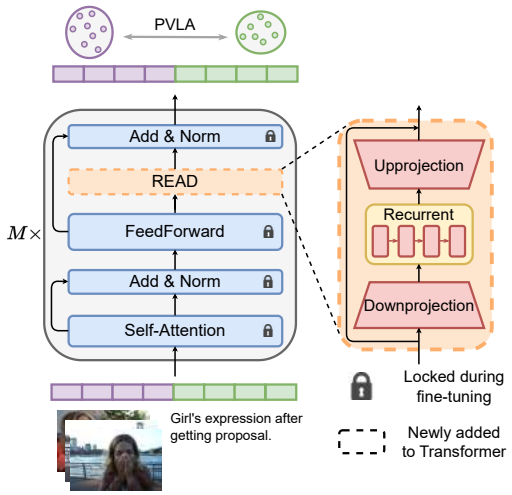

Figure 3: Overall illustration of the proposed recurrent adapter (READ) and partial video-language alignment (PVLA) framework.

benchmark but also other datasets. (Liu et al. 2022) follow with a multimodal Transformer that first considers video and language as separate input streams and unifies them, while preserving the pre-training scheme of (Lei, Berg, and Bansal 2021). For the VLS task, (Yu et al. 2021) take into account generative pre-trained Transformer to fuse the video and language content then generate the output summary.

3 Methodology

We present recurrent adapter (READ) to effectively develop the temporal modeling capability and efficiently transfer large pre-trained transformer models for video-language downstream tasks. We also introduce the partial video-language alignment (PVLA) task to optimize the alignment of in-distribution video-language inputs for better supporting video-language adaptation under low-resource settings. Our overall framework is illustrated in Figure 3.

3.1 Preliminary – Transformer Architecture for Video-Language Modeling

We concentrate our work upon the Transformer architecture (Vaswani et al. 2017). The architecture consists of an embedding layer and M consecutive Transformer blocks. As inputs to the Transformer model, we extract N_V frames and N_L words from the video and language input, respectively. The embedding layer would encode the extracted frames and words into sequences of initial video and language representations $H_V^{(0)} = \{\mathbf{h}_{v,i}^{(0)}\}_{i=1}^{N_V}$ and $H_L^{(0)} = \{\mathbf{h}_{l,j}^{(0)}\}_{j=1}^{N_L}$, respectively. The transformer then forwards these sequences into consecutive Transformer blocks, each of which is typically composed of a multi-head self-attention (MHSA) layer, a residual connection with normalization (Add & Norm) layer, a feedforward layer, and another Add & Norm layer.

In MHSA for video-language modeling, the language representations are linearly projected into the query tensor $\mathbf{Q} \in \mathbb{R}^{N_L \times d}$, whilst the video representations into the key $\mathbf{K} \in \mathbb{R}^{N_V \times d}$ and value tensors $\mathbf{V} \in \mathbb{R}^{N_V \times d}$:

$$\mathbf{Q}^{(m)} = \text{Linear}\left(\mathbf{H}_L^{(m)}\right),\ \mathbf{K}^{(m)} = \text{Linear}\left(\mathbf{H}_V^{(m)}\right),$$
$$\mathbf{V}^{(m)} = \text{Linear}\left(\mathbf{H}_V^{(m)}\right),$$

where m denotes the index of the current Transformer block and d the hidden dimension. Then, the self-attention computation is conducted upon these vectors as:

$$\mathbf{X}^{(m)} = \text{Attention}\left(\mathbf{Q}^{(m)}, \mathbf{K}^{(m)}, \mathbf{V}^{(m)}\right) = \text{Softmax}\left(\frac{\mathbf{Q}^{(m)} \cdot (\mathbf{K}^{(m)})^\top}{\sqrt{d}}\right) \cdot \mathbf{V}^{(m)}. \quad (1)$$

The attention output $\mathbf{X}^{(m)}$ is subsequently sent to an Add & Norm layer:

$$\mathbf{P}^{(m)} = \text{LN}\left(\mathbf{X}^{(m)} + \mathbf{H}_L^{(m)}\right), \quad (2)$$

where LN denotes the layer normalization layer. Subsequently, $\mathbf{P}^{(m)}$ is forwarded to a FeedForward block to produce the output representation $\mathbf{O}^{(m)}$, which will be passed to another Add & Norm layer to create the video-informed language representation for the next transformer block:

$$\mathbf{O}^{(m)} = \text{GeLU}\left(\text{Linear}\left(\mathbf{P}^{(m)}\right)\right), \quad (3)$$

$$\mathbf{H}_L^{(m+1)} = \text{LN}\left(\mathbf{P}^{(m)} + \mathbf{O}^{(m)}\right),\ \mathbf{H}_V^{(m+1)} = \mathbf{H}_V^{(m)}. \quad (4)$$

The video-language representation of the last Transformer block $\mathbf{H}_L^{(M+1)}$ is finally adopted to perform a specific downstream task.

3.2 Recurrent Adapter (READ)

The objective of our READ is to incorporate the temporal modeling capability for the adaptation module. To this end, we construct a recurrent-based bottleneck layer which is composed of a downprojection layer, a recurrent neural network (RNN) layer, and an up-projection layer.

Formally, given the FeedForward output $\mathbf{O}$, our recurrent adapter can be expressed as:

$$\tilde{\mathbf{O}} = \mathbf{O} + \text{GELU}\left(\text{RNN}\left(\mathbf{O} \cdot W_{\text{down}}\right)\right) \cdot W_{\text{up}}, \quad (5)$$

where $W_{\text{down}} \in \mathbb{R}^{d \times k}, W_{\text{up}} \in \mathbb{R}^{k \times d}$, and $k \ll d$. Subsequently, we combine $\mathbf{P}$ and $\tilde{\mathbf{O}}$ via residual connection to generate the output $\mathbf{H}$:

$$\mathbf{H} = \text{LN}\left(\tilde{\mathbf{O}} + \mathbf{P}\right). \quad (6)$$

In addition to RNN, we also experiment with other recurrent architectures in Table 8 and observe that the performance is insensitive to the architectural choice. Therefore, for simplicity, we decide to implement the RNN architecture in our READ layer.

Fine-tuning. During the fine-tuning stage, we preserve the weights of the pre-trained Transformer model and only optimize our introduced READ layers. In detail, the original model components (blue blocks in Figure 3) are frozen,

while the parameters of READ (the yellow block in Figure 3) are updated with respect to the task-specific and the partial video-language alignment losses, which will be delineated in the upcoming section.

Testing. During testing, we maintain the shared parameters of the pre-trained Transformer model and only load those of our extra READ modules that are fine-tuned in the previous phase. This would keep the storage cost from burgeoning because the number of added parameters is tiny.

3.3 Partial Video-Language Alignment (PVLA)

To encourage the control towards the information flow of video frames and language words, we propose to optimize the alignment between the in-distribution video and language representations H_V and H_L at all Transformer blocks.

We consider video and language as two discrete distributions $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, whose H_V and H_L are their supports, respectively. We formulate this setting as $\boldsymbol{\mu} = \sum_{i=1}^{N_V} \mathbf{a}_i \delta_{\mathbf{h}_{v,i}}$ and $\boldsymbol{\nu} = \sum_{j=1}^{N_L} \mathbf{b}_j \delta_{\mathbf{h}_{l,j}}$, with $\delta_{\mathbf{h}_{l,j}}$ and $\delta_{\mathbf{h}_{l,j}}$ being the Dirac functions respectively centered upon $\mathbf{h}_{v,i}$ and $\mathbf{h}_{l,j}$. The weight vector of the supports is $\mathbf{a} = \frac{\mathbf{1}_{N_V}}{N_V}$, and $\mathbf{b} = \frac{\mathbf{1}_{N_L}}{N_L}$.

Based upon the above setting, we propose the partial video-language alignment (PVLA) task, which is to minimize the following $\mathcal{L}_{\text{PVLA}}$ loss equal to the partial optimal transport (POT) distance $\mathcal{D}_{\text{POT}}$ between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ as:

$$\mathcal{L}_{\text{PVLA}} = \mathcal{D}_{\text{POT}}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\mathbf{T} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i=1}^{N_V} \sum_{j=1}^{N_L} \mathbf{T}_{i,j} \cdot c\left(\mathbf{h}_{v,i}, \mathbf{h}_{l,j}\right), \tag{7}$$

$$\text{s.t} \quad \Pi(\mathbf{a}, \mathbf{b}) = \{\mathbf{T} \in \mathbb{R}_+^{N_V \times N_L} \mid \mathbf{T}\mathbf{1}_{N_L} \leq \mathbf{a}, \mathbf{T}^\top \mathbf{1}_{N_V} \leq \mathbf{b}, \\ \mathbf{1}_{N_V}^\top \cdot \mathbf{T} \cdot \mathbf{1}_{N_L} = s, \quad 0 \leq s \leq \min(N_L, N_V)\}. \tag{8}$$

Because the exact minimization over the transport plan $\mathbf{T}$ is intractable, we adopt the Sinkhorn-based algorithm to compute $\mathbf{T}$. We explicate our algorithm to calculate the partial video-language alignment loss in Algorithm 1.

Our PVLA formulation is flexible where it allows only s samples from one distribution to be transported to the other, and enables the algorithm to decide the value of s, in case the input language only corresponds to certain video aspects (or vice versa).

Training Strategy. For training, we jointly optimize the video-language task-specific loss and our PVLA loss. It is worth noting that we only update our introduced READ layers while keeping the remaining components frozen.

Algorithm 1: Computing the PVLA loss

Require: $\mathbf{C} = \{\mathbf{C}_{i,j} = c\left(\mathbf{h}_{v,i}, \mathbf{h}_{l,j}\right) \mid 1 \leq i \leq N_V, 1 \leq j \leq N_L\} \in \mathbb{R}^{N_V \times N_L}$, temperature τ, $\mathbf{a} \in \mathbb{R}^{N_V}$, $\mathbf{b} \in \mathbb{R}^{N_L}$, s, N_{iter}

$\mathcal{L}_{\text{PVLA}} = \infty$

for $s = 1$ to $\min(N_L, N_V)$ **do**

  $\mathbf{T} = \exp\left(-\frac{\mathbf{C}}{\tau}\right)$

  $\mathbf{T} = \frac{s}{\left(\mathbf{1}_{N_V}\right)^\top \cdot \mathbf{T} \cdot \mathbf{1}_{N_L}} \mathbf{T}$

  for $i = 1$ to N_{iter} **do**

    $\mathbf{p}_a = \min\left(\frac{\mathbf{a}}{\mathbf{T}\mathbf{1}_{N_L}}, \mathbf{1}_{N_V}\right)$

    $\mathbf{T}_a = \text{diag}\left(\mathbf{p}_a\right) \cdot \mathbf{T}$

    $\mathbf{p}_b = \min\left(\frac{\mathbf{b}}{\mathbf{T}_a^\top \mathbf{1}_{N_V}}, \mathbf{1}_{N_L}\right)$

    $\mathbf{T}_b = \text{diag}\left(\mathbf{p}_b\right) \cdot \mathbf{T}_a$

    $\mathbf{T} = \frac{s}{\left(\mathbf{1}_{N_V}\right)^\top \cdot \mathbf{T} \cdot \mathbf{1}_{N_L}} \mathbf{T}_b$

  end for

  $\mathcal{L}_{\text{PVLA}} = \min\left(\mathcal{L}_{\text{PVLA}}, \sum_{i=1}^{N_V} \sum_{j=1}^{N_L} \mathbf{T}_{i,j} \mathbf{C}_{i,j}\right)$

end for

return $\mathcal{L}_{\text{PVLA}}$

4 Experiments

We conduct extensive experiments to evaluate the effectiveness of our READ-PVLA framework. We first describe the experimental settings, covering the downstream tasks, evaluation metrics, pre-trained backbones, baseline approaches, and implementation details. We then present the numerical results of our method with baseline models, then provide ablation study and thorough analysis to explore various configurations. Eventually, we perform qualitative assessments to further elucidate the behavior of our framework.

4.1 Experimental Settings

Downstream tasks. We assess the effectiveness on the temporal language grounding and video-language summarization tasks. The corresponding datasets to each task are presented as follows:

- *Temporal Language Grounding (TLG):* The TLG's task is to localize temporal boundaries of the video frames that semantically relate to the language query. The evaluation is performed upon three datasets, *i.e.* YouTube Highlights (Sun, Farhadi, and Seitz 2014), TVSum (Song et al. 2015), and QVHighlights (Lei, Berg, and Bansal 2021). YouTube Highlights consists of 40 video-language training inputs for each of the 6 domains. TVSum comprises 10 domains, each of which possesses 5 video-language training inputs. The QVHighlights benchmark includes 7,218 language-annotated video segments for training, 1,550 for development, and 1,542 for testing. Following previous work on low-resource experiments (Boulanger, Lavergne, and Rosset 2022), we keep our training size at 700 samples, which is less than 10% of the full data for the QVHighlights dataset, while preserving the original splits on the TVSum and YouTube Highlights datasets.
- *Video-Language Summarization (VLG):* Given a video-language input, the VLS's target is to generate a summary which takes into account both video and language content (Yu et al. 2021). We consider the How2 dataset (Sanabria et al. 2018), from which we randomly draw 2,000 out of 73,993 samples for training, *i.e.* less than

Method	#params (M)	Dog	Gym	Par.	Ska.	Ski.	Sur.	Avg.
Full	283.97 (100%)	65.90‡	75.20‡	82.20‡	71.80‡	72.30‡	81.15‡	74.76‡
Bias	0.51 (0.18%)	46.23‡	61.19‡	56.73‡	31.36‡	61.14‡	49.77‡	51.07‡
Partial	38.75 (13.65%)	48.28‡	63.26‡	59.71‡	32.66‡	64.58‡	56.22‡	54.12‡
Proj	5e-4 (1.76e-4%)	57.05‡	65.70‡	63.03‡	71.83‡	65.45‡	79.71‡	67.13‡
LoRA	13.12 (4.62%)	60.97‡	67.68‡	72.53‡	66.62‡	71.24‡	79.15‡	69.70‡
Prompt	0.02 (0.01%)	48.28‡	63.26‡	59.71‡	35.67‡	35.67‡	64.61‡	46.87‡
Adapter	13.11 (4.62%)	62.89‡	67.09‡	74.56‡	62.56‡	68.10‡	78.73‡	68.98‡
READ-PVLA	0.16 (0.06%)	**67.65**	**78.05**	**83.25**	**72.40**	**72.98**	**82.36**	**76.12**

Table 1: TLG results on the YouTube Highlights dataset. We report the mean average precision (mAP) and the number of trainable parameters (#params). ‡means the gain of READ-PVLA is statistically significant at the 0.05 level.

Method	#params (M)	VT	VU	GA	MS	PK	PR	FM	BK	BT	DS	Avg.
Full	285.28 (100%)	84.17‡	81.50‡	88.20‡	71.54‡	81.40‡	84.31‡	72.30‡	76.53‡	78.86‡	77.70‡	79.65‡
Bias	0.25 (0.09%)	38.08‡	69.62‡	60.87‡	31.25‡	68.84‡	51.71‡	50.72‡	65.38‡	54.42‡	59.05‡	54.99‡
Partial	38.75 (13.58%)	57.27‡	62.57‡	58.08‡	52.35‡	61.58‡	63.94‡	50.82‡	62.36‡	58.05‡	47.79‡	57.48‡
Proj	5e-4 (1.75e-4%)	57.65‡	65.80‡	64.40‡	55.57‡	64.67‡	67.07‡	59.08‡	74.70‡	63.29‡	49.48‡	62.17‡
LoRA	13.28 (4.66%)	77.87‡	77.01‡	77.82‡	66.38‡	80.21‡	82.23‡	66.89‡	72.31‡	69.58‡	72.09‡	74.24‡
Prompt	0.02 (0.007%)	61.67‡	71.98‡	64.07‡	35.54‡	72.74‡	48.70‡	52.97‡	67.59‡	57.28‡	38.60‡	57.11‡
Adapter	13.29 (4.66%)	78.46‡	76.38‡	77.36‡	67.12‡	80.33‡	82.51‡	67.77‡	71.71‡	69.58‡	71.24‡	74.25‡
READ-PVLA	0.14 (0.05%)	**88.30**	**85.15**	**89.76**	**75.80**	**86.69**	**86.62**	**74.99**	**82.38**	**84.65**	**79.60**	**83.39**

Table 2: TLG results on the TVSum dataset. We report the mean average precision (mAP) and the number of trainable parameters (#params). ‡means the gain of READ-PVLA is statistically significant at the 0.05 level.

Method	#params (M)	mAP
Full	15.88 (100%)	36.14‡
Bias	0.06 (0.38%)	24.89‡
Partial	1.05 (6.61%)	26.37‡
Proj	7.31 (46.03%)	32.71‡
LoRA	0.19 (1.20%)	33.96‡
Prompt	0.04 (0.25%)	25.86‡
Adapter	0.20 (1.26%)	33.61‡
READ-PVLA	0.19 (1.20%)	**36.74**

Table 3: TLG results on the QVHighlights dataset. We report the mean average precision (mAP) and the number of trainable parameters (#params). ‡means the gain of READ-PVLA is statistically significant at the 0.05 level.

3% of the full data, to simulate the low-resource settings, while maintaining 2,520 samples for validation, and 2,127 samples for testing.

Evaluation metrics. For the TLG task, we follow previous works (Lei, Berg, and Bansal 2021; Liu et al. 2022) to use the mean average precision (mAP) metric. Regarding VLS, we utilize the ROUGE score, which is a popular metric for summarization (Zhang et al. 2020; Yu et al. 2021).

Pre-trained backbones. We adopt the Transformer encoder-decoder architecture (Vaswani et al. 2017) pre-trained with both supervised and self-supervised objectives. Specifically, for TLG, we use the unified multimodal transformer (UMT) (Liu et al. 2022) and Moment-DETR (Lei, Berg, and Bansal 2021) models pre-trained upon the automatic speech recognition task. For VLS, we carry out the parameter-efficient adaptation on the generative vision-guided BART (VG-BART) and T5 (VG-T5) (Yu

Method	#params (M)	R1	R2	RL
Full	249.67 (100%)	35.72‡	11.88‡	30.00‡
Bias	0.20 (0.08%)	30.51‡	8.20‡	23.00‡
Partial	16.54 (6.62%)	31.55‡	8.63‡	13.65‡
Proj	38.60 (15.46%)	32.76‡	9.11‡	30.01‡
LoRA	1.19 (0.48%)	40.04‡	20.36‡	35.98‡
Prompt	0.05 (0.02%)	31.55‡	8.63‡	14.90‡
Adapter	1.20 (0.48%)	41.52‡	20.75‡	36.88‡
READ-PVLA	1.17 (0.47%)	**44.01**	**21.91**	**37.91**

Table 4: VLS results on the How2 dataset with the VG-BART model. We report the ROUGE-1, ROUGE-2, and ROUGE-L scores, with the number of trainable parameters (#params). ‡means the gain of READ-PVLA is statistically significant at the 0.05 level.

Method	#params (M)	R1	R2	RL
Full	333.16 (100%)	32.37‡	8.07‡	26.53‡
Bias	0.07 (0.02%)	27.03‡	4.53‡	19.51‡
Partial	16.52 (4.96%)	27.83‡	4.92‡	10.49‡
Proj	24.67 (7.40%)	29.16‡	5.50‡	26.76‡
LoRA	1.93 (0.58%)	36.63‡	16.72‡	32.77‡
Prompt	0.18 (0.05%)	28.33‡	5.12‡	11.75‡
Adapter	1.95 (0.59%)	37.83‡	17.45‡	33.87‡
READ-PVLA	1.91 (0.57%)	**40.12**	**18.71**	**34.42**

Table 5: VLS results on the How2 dataset with the VG-T5 model. We report the ROUGE-1, ROUGE-2, and ROUGE-L scores, with the number of trainable parameters (#params). ‡means the gain of READ-PVLA is statistically significant at the 0.05 level.

Method	mAP - YouTube Highlights	R2 - How2
No VLA	73.80	18.22
VLA	74.41	20.01
PVLA	76.12	21.91

Table 6: Partial video-language alignment (PVLA) ablation experiments on YouTube Highlights and How2. We color the settings we implement for our READ-PVLA method.

et al. 2021) pre-trained upon reconstruction and masked language modeling tasks (Raffel et al. 2020).

Baseline methods. We compare our method with a comprehensive list of baseline approaches for efficient video-language transfer learning:

- *Full*: update all parameters of the pre-trained backbone.
- *Partial*: only update the last layers of the encoder and decoder in the Transformer model.
- *Bias* (Zaken, Goldberg, and Ravfogel 2022): only fine-tune the bias terms in the Transformer backbone.
- *Proj*: fine-tune only the last linear projection layer in the Transformer.
- *LoRA* (Hu et al. 2021): solely fine-tune the decomposition matrices introduced to the linear weights of the Transformer model.
- *Prompt* (Jia et al. 2022): Append a sequence of learnable prompt tokens to both video and language inputs and only fine-tune the appended sequence.
- *Adapter* (Houlsby et al. 2019): update only the adaptation modules consisting of downprojection and up-projection layers inserted into the Transformer model.

Implementation details. For the TLG task, we use the SlowFast (Feichtenhofer et al. 2019) and video encoder of CLIP (Radford et al. 2021) to extract features every 2 seconds. For the VLS task, we use a 3D ResNeXt-101 model to extract a 2048-dimensional embedding for every 16 non-overlapping frames. Similar to previous works (Houlsby et al. 2019; Chen et al. 2022), to support training stability, we initialize the weights of the down-projection layer W_{down} with the Kaiming normal (He et al. 2015) method, whereas those of the up-projection W_{up}, recurrent layer RNN, and biases of our READ layers are configured with zero initialization. In our PVLA framework, we implement the cost distance $c(\mathbf{h}_{v,i}, \mathbf{h}_{l,j})$ as the cosine distance $c(\mathbf{h}_{v,i}, \mathbf{h}_{l,j}) = 1 - \frac{\mathbf{h}_{v,i} \cdot \mathbf{h}_{l,j}}{||\mathbf{h}_{v,i}||_2 \cdot ||\mathbf{h}_{l,j}||_2}$, and set the maximum number of iterations N_{iter} to 1,000 and the temperature τ to 0.05. We fine-tune all models leveraging the AdamW optimizer on 4 NVIDIA Tesla V100 GPUs and report average results of 5 runs. Specific details about the epoch, batch size, learning rate, and the number of Transformer blocks for each task can be found in the Appendix.

4.2 Main Results

For main comparison of our READ-PVLA with baseline methods, we denote the results of YouTube Highlights in Table 1, TVSum in Table 2, QVHighlights in Table 3, and How2 in Tables 4 and 5.

Temporal language grounding (TLG). For the YouTube Highlights dataset, our READ-PVLA framework substantially outperforms the Full fine-tuning approach (*e.g.* 1.36% on average, 1.75% in the Dog domain, and 1.21% in the Surfing domain), while updating far less parameters (0.16M vs. 283.97M). We significantly surpass all other efficient fine-tuning methods as well, *e.g.* with an improvement of 10.96% over the Adapter in the Gym category, or 5.78% over LoRA in the Skating category.

For the TVSum dataset, we observe that our method enhances the Full fine-tuning direction with only 0.14M versus 285.28M tunable parameters. For instance, we obtain an increase of 4.26% in the MS subset and 3.74% on average. Compared with the best parameter-efficient approach, *i.e.* the Adapter, we achieve a gain of 15.07% in BT, 10.67% in BK, and 8.77% in the VU domain.

Our improvement also generalizes across different pre-trained backbone. On the QVHighlights dataset, in which we work with the Moment-DETR architecture, we accomplish a gain of 0.6% over the standard fine-tuning method, while our tunable parameters are only 0.19M versus its 15.88M. We also surpass the efficient approach LoRA with an enhancement of 2.78% in mAP.

These results demonstrate that our READ-PVLA framework can efficiently model video-language inputs to polish the low-resource temporal language grounding performance of various pre-trained Transformer models.

Video-language summarization (VLS). Analogous to the TLG experiment, on the VG-BART backbone, we improve upon the full fine-tuning approach with 8.29 points of ROUGE-1, 10.03 points of ROUGE-2, and 7.91 points of ROUGE-L. Importantly, we only update 1.17M parameters, which account for 0.47% total parameters of the overall model. On the VG-T5 backbone, we exceed the full approach by 7.75 points in ROUGE-1, 10.64 points in ROUGE-2, and 7.89 points in ROUGE-L, whilst keeping 99.43% parameters frozen.

In addition, our framework substantially outperforms other fine-tuning strategies, *e.g.* LoRA with 3.97 points in ROUGE-1, 1.55 points in ROUGE-2, and 1.93 points in ROUGE-L on the VG-BART architecture, along with 3.49 points in ROUGE-1, 1.99 points in ROUGE-2, and 1.65 points in ROUGE-L on the VG-T5 one.

These results substantiate that our method is applicable to diverse benchmarks and model architectures, particularly not only multimodal Transformers for temporal language grounding but also generative Transformers for video-language summarization. We hypothesize that our advantages are due to the recurrent adapter's ability to model temporal information and the PVLA task to align video and language signals to maintain more essential information during the efficient fine-tuning stage.

4.3 Ablation Studies

We ablate our READ-PVLA framework to discover what factors result in the demonstrated efficiency and observe several intriguing properties. Our ablation studies are all conducted on the YouTube Highlights and How2 test set.

Video	Language query	POT distance	AP
	Girl's expression after getting proposal	27.17	90.28
	A boy showing his arm after being stung at the beach	56.38	30.00

Table 7: Case study on the temporal language grounding benchmark. We extract the POT distance between video and language of two inputs with different language queries and measure the respective AP performance change. The video flows from top to bottom and left to right.

Recurrent architecture	#params - UMT (M)	mAP - YouTube Highlights	#params - VG-BART (M)	R2 - How2
GRU	0.16	76.07	1.17	21.85
LSTM	0.16	76.08	1.17	21.87
RNN	0.16	76.12	1.17	21.91

Table 8: Recurrent architecture ablation experiment on YouTube Highlights and How2. We color the settings we implement for our READ-PVLA method.

Distance method	mAP - YouTube Highlights	R2 - How2
AvgPool - Cosine	71.65	20.32
MaxPool - Cosine	74.37	21.36
AvgPool - L2	72.11	20.73
MaxPool - L2	74.39	21.02
Partial OT	76.12	21.91

Table 9: Distance method ablation experiments on YouTube Highlights and How2. We color the settings we implement for our READ-PVLA method.

Effects of video-language alignment. We evaluate our framework without the assistance of the PVLA task and with the one of the VLA variant that requires all masses of one distribution to be transferred (we set $s = \min(N_V, N_L)$ in formulation (8)). As shown in Table 6, the performance drops dramatically when we remove the PVLA task from the fine-tuning procedure. We conjecture that the model has become deficient in managing the information injected into the low-dimensional space of the READ layers, thus passing detrimental noise to the downstream task. Moreover, the VLA variant brings slight performance decrease, which could be due to the VLA's restrictive nature of transporting all masses from the language distribution to the video one or vice versa.

Effects of the recurrent architecture. In addition to RNN, there exist various recurrent architectures in the literature, particularly the gated recurrent unit (GRU) (Cho et al. 2014) and long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997). We experiment with different recurrent choices and explicate the results in Table 8. As can be observed, the performance is insensitive to the choice of recurrent design. Therefore, we select the simplest option, *i.e.* recurrent neural network (RNN) for our READ layers.

Distance methods. We further ablate on the distance metrics to estimate the distance between video and language distributions. Technically, we perform the average- and max-pooling of the video and language representations. Then, we consider the cosine distance or the L2 distance of the two pooled vectors as the video-language distance. Results in Table 9 substantiate the superiority of our POT distance for the PVLA objective. Such success illustrates the POT-based PVLA's advantage of modeling the relationship nature between video and language representations

4.4 Qualitative Assessment

Case study. We display a TLG example on the YouTube Highlights dataset, along with the POT distance estimated by our PVLA framework and the AP score in Table 7. We observe that when the language query semantically corresponds to a moment in the video, *i.e. a girl expression after she gets the proposal*, the POT distance is small and correlates with the high value of AP. In contrast, when we replace the original query with an out-of-distribution one, the POT distance burgeons significantly, causing the AP to decrease from 90.28% to 30.00%. Therefore, we conclude that our READ-PVLA framework is capable of intelligently adjusting the information flowing through the READ layers in order to produce the final output consistent with the video-language input and downstream tasks.

5 Conclusion

We propose a novel READ-PVLA framework for parameter-efficient transfer learning to video-language modeling tasks. Our READ-PVLA utilizes recurrent computation component to enable temporal modeling capability and partial video-language alignment objective to preserve critical information for bottleneck adaptation modules. Experiments demonstrate that READ-PVLA consistently outperforms both the full fine-tuning and competitive strategies, whilst bringing the benefit of parameter-efficiency (at most 1.20% trainable parameters). Our method is also applicable to diverse pre-trained models, which has the potential to employ more powerful video-language models in the future.

Acknowledgements

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG3-PhD-2023-08-051T). Thong Nguyen is supported by a Google Ph.D. Fellowship in Natural Language Processing.

References

Boulanger, H.; Lavergne, T.; and Rosset, S. 2022. Generating unlabelled data for a tri-training approach in a low resourced NER task. In *Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, 30–37. Association for Computational Linguistics.

Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; and Luo, P. 2022. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35: 16664–16678.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6202–6211.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*. MIT press.

Han, W.; Chen, H.; and Poria, S. 2021. Improving multimodal fusion with hierarchical mutual information maximization for multimodal sentiment analysis. *arXiv preprint arXiv:2109.00412*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2790–2799. PMLR.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, S.; Gong, B.; Pan, Y.; Jiang, J.; Lv, Y.; Li, Y.; and Wang, D. 2023. VoP: Text-Video Co-operative Prompt Tuning for Cross-Modal Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6565–6574.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *European Conference on Computer Vision*, 709–727. Springer.

Jiang, H.; Zhang, J.; Huang, R.; Ge, C.; Ni, Z.; Lu, J.; Zhou, J.; Song, S.; and Huang, G. 2022. Cross-modal adapter for text-video retrieval. *arXiv preprint arXiv:2211.09623*.

Lei, J.; Berg, T. L.; and Bansal, M. 2021. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34: 11846–11858.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880.

Liu, S.; Cao, J.; Yang, R.; and Wen, Z. 2023. Long Text and Multi-Table Summarization: Dataset and Method. *arXiv preprint arXiv:2302.03815*.

Liu, Y.; Li, S.; Wu, Y.; Chen, C.-W.; Shan, Y.; and Qie, X. 2022. Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3042–3051.

Nguyen, T.; Wu, X.; Dong, X.; Nguyen, C.-D.; Ng, S.-K.; and Tuan, L. A. 2023. DemaFormer: Damped Exponential Moving Average Transformer with Energy-Based Modeling for Temporal Language Grounding. *arXiv preprint arXiv:2312.02549*.

Pan, J.; Lin, Z.; Zhu, X.; Shao, J.; and Li, H. 2022. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35: 26462–26477.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Sanabria, R.; Caglayan, O.; Palaskar, S.; Elliott, D.; Barrault, L.; Specia, L.; and Metze, F. 2018. How2: a large-scale dataset for multimodal language understanding. *arXiv preprint arXiv:1811.00347*.

Song, Y.; Vallmitjana, J.; Stent, A.; and Jaimes, A. 2015. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5179–5187.

Sun, M.; Farhadi, A.; and Seitz, S. 2014. Ranking domain-specific highlights by analyzing edited videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, 787–802. Springer.

Sung, Y.-L.; Cho, J.; and Bansal, M. 2022. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5227–5237.

Tsai, Y.-H. H.; Ma, M. Q.; Yang, M.; Salakhutdinov, R.; and Morency, L.-P. 2020. Multimodal routing: Improving local and global interpretability of multimodal language analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2020, 1823. NIH Public Access.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xu, M.; Zhang, Z.; Wei, F.; Hu, H.; and Bai, X. 2023. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2945–2954.

Yang, T.; Zhu, Y.; Xie, Y.; Zhang, A.; Chen, C.; and Li, M. 2023. Aim: Adapting image models for efficient video action recognition. *arXiv preprint arXiv:2302.03024*.

Yu, T.; Dai, W.; Liu, Z.; and Fung, P. 2021. Vision Guided Generative Pre-trained Language Models for Multimodal Abstractive Summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3995–4007.

Zaken, E. B.; Goldberg, Y.; and Ravfogel, S. 2022. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 1–9.

Zhang, B.; Jin, X.; Gong, W.; Xu, K.; Zhang, Z.; Wang, P.; Shen, X.; and Feng, J. 2023. Multimodal video adapter for parameter efficient video text retrieval. *arXiv preprint arXiv:2301.07868*.

Zhang, J.; Zhao, Y.; Saleh, M.; and Liu, P. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, 11328–11339. PMLR.