HERO: Hierarchical Encoder for Video+Language Omni-representation Pre-training

Linjie Li[*], **Yen-Chun Chen**[*], **Yu Cheng**, **Zhe Gan**, **Licheng Yu**, **Jingjing Liu**
Microsoft Dynamics 365 AI Research
{lindsey.li, yen-chun.chen, yu.cheng, zhe.gan, licheng.yu, jingjl}@microsoft.com

Abstract

We present HERO, a novel framework for large-scale video+language omni-representation learning. HERO encodes multimodal inputs in a hierarchical structure, where *local* context of a video frame is captured by a Cross-modal Transformer via multimodal fusion, and *global* video context is captured by a Temporal Transformer. In addition to standard Masked Language Modeling (MLM) and Masked Frame Modeling (MFM) objectives, we design two new pre-training tasks: (*i*) Video-Subtitle Matching (VSM), where the model predicts both global and local temporal alignment; and (*ii*) Frame Order Modeling (FOM), where the model predicts the right order of shuffled video frames. HERO is jointly trained on HowTo100M and large-scale TV datasets to gain deep understanding of complex social dynamics with multi-character interactions. Comprehensive experiments demonstrate that HERO achieves new state of the art on multiple benchmarks over Text-based Video/Video-moment Retrieval, Video Question Answering (QA), Video-and-language Inference and Video Captioning tasks across different domains. We also introduce two new challenging benchmarks How2QA and How2R for Video QA and Retrieval, collected from diverse video content over multimodalities.[1]

1 Introduction

Inspired by BERT (Devlin et al., 2019), large-scale multimodal pre-training has prevailed in the realm of vision-and-language research (Lu et al., 2019; Tan and Bansal, 2019; Chen et al., 2020b). There are many early players in the area, including ViLBERT (Lu et al., 2019), LXMERT (Tan and Bansal, 2019), UNITER (Chen et al., 2020b), VL-BERT (Su et al., 2020) and Unicoder-VL (Li et al., 2020a). However, most large-scale pre-trained models are tailored for static images, not dynamic videos. VideoBERT (Sun et al., 2019b) is the first to apply BERT to learn joint embedding for video-text pairs. But since only discrete tokens are used to represent video frames, rich video frame features are not fully utilized. To remedy this, CBT (Sun et al., 2019a) proposes to use a contrastive loss, but mainly for video representation learning alone, with text input only considered as side information. UniViLM (Luo et al., 2020) takes a step further and considers both understanding and generation tasks.

Several constraints inherently limit the success of existing models. (*i*) Most model designs are direct adaptation of BERT, taking simple concatenation of subtitle sentences and visual frames as input, while losing the temporal alignment between video and text modalities. (*ii*) Pre-training tasks are directly borrowed from image+text pre-training methods, without exploiting the sequential nature of videos. (*iii*) Compared to diverse image domains investigated in existing work, video datasets used in current models are restricted to cooking or narrated instructional videos (Miech et al., 2019), excluding video sources that contain dynamic scenes and complex social interactions.

To tackle these challenges, we present a new video-and-language large-scale pre-training framework - HERO (**H**ierarchical **E**ncode**R** for **O**mni-representation learning). As illustrated in Figure 1, HERO takes as input a sequence of video clip frames and their accompanying subtitle sentences.[2] Instead of adopting a flat BERT-like encoder, HERO encodes multimodal inputs in a hierarchical fashion, with (*i*) a *Cross-modal* Transformer to fuse a subtitle sentence and its accompanying local video

[*] Equal contribution.

[1] Code and new datasets will be released at https://github.com/linjieli222/HERO.

[2] ASR can be applied when subtitles are unavailable.

frames, followed by (ii) a *Temporal* Transformer to obtain a sequentially contextualized embedding for each video frame, using all the surrounding frames as global context. The proposed hierarchical model first absorbs visual and textual local context on frame level, which is then transferred to a global video-level temporal context. Experiments show that this novel model design achieves better performance than a flat BERT-like architecture.

Four pre-training tasks are designed for HERO: (i) Masked Language Modeling (MLM); (ii) Masked Frame Modeling (MFM); (iii) Video-Subtitle Matching (VSM); and (iv) Frame Order Modeling (FOM). Compared to prior work, the key novelty is VSM and FOM, which encourage explicit temporal alignment between multimodalities as well as full-scale exploitation of the sequential nature of video input. In VSM, the model considers not only global alignment (predicting whether a subtitle matches the input video clip), but also local temporal alignment (retrieving the moment where the subtitle should be localized in the video clip). In FOM, we randomly select and shuffle a subset of video frames, and the model is trained to restore their original order. Extensive ablation studies demonstrate that both VSM and FOM play a critical role in video+language pre-training.

To empower the model with richer knowledge beyond instructional videos used in prior work, we jointly train HERO with both HowTo100M (narrated instructional videos) (Miech et al., 2019) and a large-scale TV dataset (containing TV episodes spanning across different genres) (Lei et al., 2018, 2020a,b; Liu et al., 2020). Compared to factual descriptions in HowTo100M, the TV dataset contains more complex plots that require comprehensive interpretation of human emotions, social dynamics and causal relations of events, making it a valuable supplement to HowTo100M and a closer approximation to real-life scenarios.

Existing pre-trained models are evaluated on YouCook2 (Zhou et al., 2018a) and MSR-VTT (Xu et al., 2016a) datasets. YouCook2 focuses on cooking videos only, and the captions in MSR-VTT are very simple. To evaluate our model on more challenging benchmarks, we collect two new datasets on video-moment retrieval and question answering, *How2R* and *How2QA*. In addition, we evaluate HERO on popular retrieval and QA tasks such as TVR (Lei et al., 2020b) and TVQA (Lei et al., 2018), where HERO outperforms existing models by a large margin. We further demonstrate the generalizability of our model by adapting it to (i) diverse downstream tasks: video-and-language inference and video captioning tasks, achieving new state of the art on VIOLIN (Liu et al., 2020) and TVC (Lei et al., 2020b) benchmarks; (ii) different video types: single-channel videos (video-only) and multi-channel videos (video + subtitle), reporting superior performance over existing state of the art on DiDeMo (Anne Hendricks et al., 2017a) and MSR-VTT.

Our main contributions are summarized as follows. (i) We present HERO, a hierarchical Transformer-based model for video+language representation learning. (ii) We propose new pre-training tasks VSM and FOM, which complement MLM and MRM objectives by better capturing temporal alignment between multimodalities in both global and local contexts. (iii) Different from previous work that mainly relies on HowTo100M, we include additional video datasets for pre-training, encouraging the model to learn from richer and more divserse visual content. (iv) We collect two new datasets based on HowTo100M for video-moment retrieval/QA, and will release the new benchmarks to foster future study. HERO achieves new state of the art across all the evaluated tasks.

2 Related Work

Since the birth of BERT (Devlin et al., 2019), there has been continuing advancement in language model pre-training, such as XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), UniLM (Dong et al., 2019), and T5 (Raffel et al., 2019), which epitomizes the superb power of large-scale pre-training. Satellited around BERT, there is parallel growing interest in model compression (Sun et al., 2019c; Shen et al., 2020) and extension to generation tasks (Chen et al., 2020a; Wang and Cho, 2019).

Branching out from language processing to multimodal, subsequent studies also emerge in vision+language space. Prominent work includes ViLBERT (Lu et al., 2019), LXMERT (Tan and Bansal, 2019), VL-BERT (Su et al., 2020), Unicoder-VL (Li et al., 2020a), B2T2 (Alberti et al., 2019), UNITER (Chen et al., 2020b) and VILLA (Gan et al., 2020). A detailed review can be found in Appendix A.7.

Contrast to the boom in image+text area, pre-training for video+language is still in its infancy.

So far, VideoBERT (Sun et al., 2019b), CBT (Sun et al., 2019a), MIL-NCE (Miech et al., 2020), ActBERT (Zhu and Yang, 2020) and UniViLM (Luo et al., 2020) are the only existing work exploring this space, covering downstream tasks from text-based video retrieval (Zhou et al., 2018a; Xu et al., 2016b) and video question answering (Maharaj et al., 2017; Lei et al., 2020a) to video captioning (Zhou et al., 2018b).

In this paper, we aim to propel video+language omni-representation learning in four dimensions: (i) better model architecture design; (ii) better pre-training task design; (iii) diversification of training corpora; and (iv) new high-quality benchmarks for downstream evaluation.

3 Hierarchical Video+Language Encoder

In this section, we explain the proposed HERO architecture and the four pre-training tasks in detail.

3.1 Model Architecture

Model architecture of HERO is illustrated in Figure 1, which takes the frames of a video clip and the textual tokens of subtitle sentences as inputs. They are fed into a Video Embedder and a Text Embedder to extract initial representations. HERO computes contextualized video embeddings in a hierarchical procedure. First, *local* textual context of each visual frame is captured by a Cross-modal Transformer, computing the contextualized multi-modal embeddings between a subtitle sentence and its associated visual frames. The encoded frame embeddings of the whole video clip are then fed into Temporal Transformer to learn the *global* video context and obtain the final contextualized video embeddings.

Input Embedder We denote visual frames of a video clip as $\mathbf{v} = \{v_i\}_{i=1}^{N_v}$ and its subtitle as $\mathbf{s} = \{s_i\}_{i=1}^{N_s}$ (N_v is the number of visual frames in a video clip and N_s is the number of sentences in each subtitle). For *Text Embedder*, we follow Liu et al. (2019) and tokenize a subtitle sentence s_i into a sequence of WordPieces (Wu et al., 2016), *i.e.*, $\mathbf{w}_{s_i} = \{w_{s_i}^j\}_{j=1}^{L}$ (L is the number of tokens in s_i). The final representation for each sub-word token is obtained via summing up its token embedding and position embedding, followed by a layer normalization (LN) layer. For *Video Embedder*, we first use ResNet (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009) and SlowFast (Feichtenhofer et al., 2019) pre-trained on Kinetics (Kay et al., 2017) to extract 2D and 3D visual features for each video frame. These features are concatenated as visual features and fed through a fully-connected (FC) layer to be projected into the same lower-dimensional space as token embeddings. Since video frames are sequential, their position embeddings can be calculated in the same way as in the Text Embedder. The final embedding of a frame is obtained by summing up FC outputs and position embeddings and then passing through an LN layer. After Input Embedder, token and frame embeddings for $\mathbf{w}_{s_i}$ and $\mathbf{v}_{s_i}$[3] are denoted as $\mathbf{W}_{s_i}^{emb} \in \mathbb{R}^{L\times d}$ and $\mathbf{V}_{s_i}^{emb} \in \mathbb{R}^{K\times d}$ (d is the hidden size).

Cross-modal Transformer To utilize the inherent alignment between subtitles and video frames, for each subtitle sentence s_i, we first learn contextualized embeddings between the corresponding tokens $\mathbf{w}_{s_i}$ and its associated visual frames $\mathbf{v}_{s_i}$ through cross-modal attention. Inspired by the recent success (Chen et al., 2020b; Lu et al., 2019) of using Transformer (Vaswani et al., 2017) for multimodal fusion, we also use a multi-layer Transformer here. The outputs from Cross-modal Transformer is a sequence of contextualized embeddings for each subtitle token and each video frame:

$$\mathbf{V}_{s_i}^{cross}, \mathbf{W}_{s_i}^{cross} = f_{cross}(\mathbf{V}_{s_i}^{emb}, \mathbf{W}_{s_i}^{emb}), \quad (1)$$

where $f_{cross}(\cdot,\cdot)$ denotes the Cross-modal Transformer, $\mathbf{V}_{s_i}^{cross} \in \mathbb{R}^{K\times d}$ and $\mathbf{W}_{s_i}^{cross} \in \mathbb{R}^{L\times d}$.

Temporal Transformer After collecting all the visual frame embeddings $\mathbf{V}^{cross} = \{\mathbf{V}_{s_i}^{cross}\}_{i=1}^{N_s} \in \mathbb{R}^{N_v\times d}$ from the output of Cross-modal Transformer, we use another Transformer as temporal attention to learn contextualized video embeddings from the global context of a video clip. To avoid losing positional information, we use residual connection (He et al., 2016) to add back $\mathbf{V}^{emb} \in \mathbb{R}^{N_v\times d}$. The final contextualized video embeddings are calculated as:

$$\mathbf{V}^{temp} = f_{temp}(\mathbf{V}^{emb} + \mathbf{V}^{cross}), \quad (2)$$

where $f_{temp}(\cdot)$ denotes the Temporal Transformer, and $\mathbf{V}^{temp} \in \mathbb{R}^{N_v\times d}$. Compared to flat BERT-like encoder, which directly concatenates all textual tokens and visual frames as inputs, the proposed

[3] $\mathbf{v}_{s_i} = \{v_{s_i}^j\}_{j=1}^{K}$ denotes the set of visual frames paired with subtitle sentence s_i, based on their timestamps. Refer to Appendix A.4 for details.

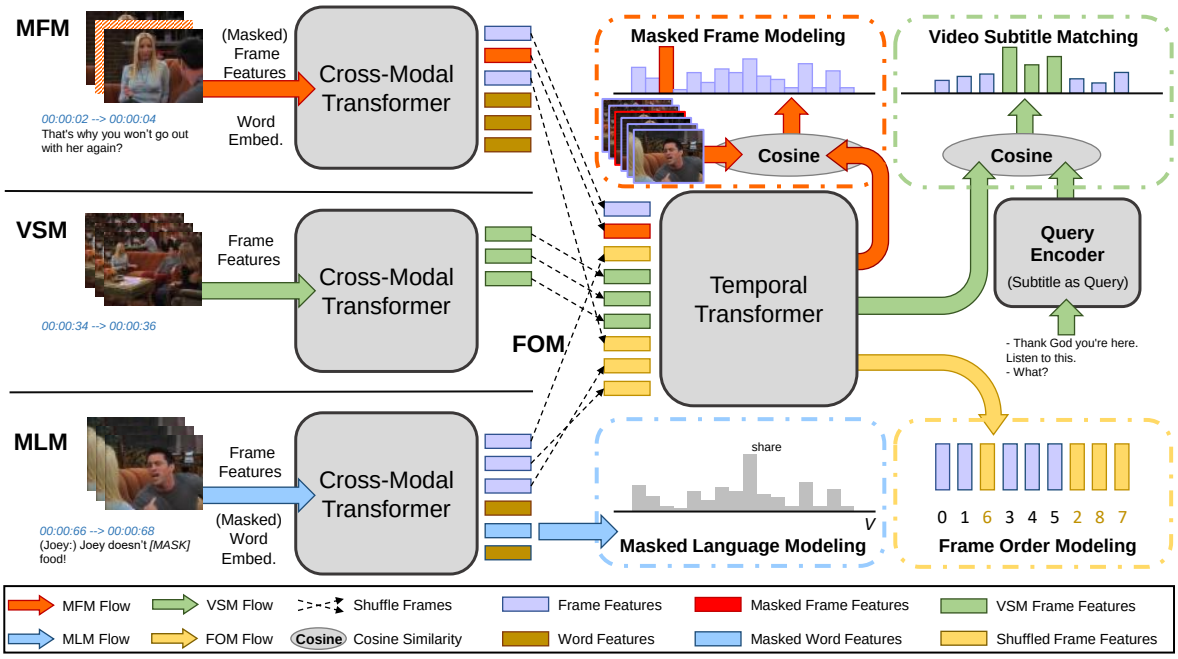


Figure 1: HERO Architecture (best viewed in color), consisting of Cross-Modal Transformer and Temporal Transformer, learned via four pre-training tasks hierarchically. Initial frame features are obtained by SlowFast and ResNet feature extractors, and word embeddings are learned via an embedding layer initialized from RoBERTa.

model effectively utilizes the temporal alignment between subtitle sentences and video frames for multimodal fusion in a more fine-grained manner. In the experiments, we show that our model design far outperforms a flat BERT-like baseline.

3.2 Pre-training Tasks

We introduce four tasks for pre-training. During training, we sample one task per mini-batch to prevent different tasks from corrupting each others' input. As shown in Figure 1, MFM and MLM are in analogy to BERT (Devlin et al., 2019). Word masking is realized by replacing a word with special token `[MASK]`, and frame masking by replacing a frame feature vector with zeros. Following Chen et al. (2020b), we only mask one modality each time while keeping the other modality intact. VSM is designed to learn both *local* alignment (between visual frames and a subtitle sentence) and *global* alignment (between a video clip and a sequence of subtitle sentences). FOM is designed to model sequential characteristics of video, by learning the original order of randomly reordered frames.

3.2.1 Masked Language Modeling

The inputs for MLM include: (i) sub-word tokens from the i-th subtitle sentence $\mathbf{w}_{s_i}$; (ii) visual frames $\mathbf{v}_{s_i}$ aligned with $\mathbf{w}_{s_i}$; and (iii) mask indices $\mathbf{m} \in \mathbb{N}^M$.[4]

In MLM, we randomly mask out input words with a probability of 15%, and replace the masked tokens $\mathbf{w}_{s_i}^{\mathbf{m}}$ with special tokens `[MASK]`.[5] The goal is to predict these masked words based on the observation of their surrounding words $\mathbf{w}_{s_i}^{\backslash \mathbf{m}}$ and the visual frames aligned with the sentence $\mathbf{v}_{s_i}$, by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{MLM}}(\theta) = -\mathbb{E}_D \log P_\theta(\mathbf{w}_{s_i}^{\mathbf{m}} | \mathbf{w}_{s_i}^{\backslash \mathbf{m}}, \mathbf{v}_{s_i}) \,, \quad (3)$$

where θ denotes trainable parameters. Each pair $(\mathbf{w}_{s_i}, \mathbf{v}_{s_i})$ is sampled from the training set D.

3.2.2 Masked Frame Modeling

Similar to MLM, we also sample frames and mask their visual features with a probability of 15%. However, the difference is that MLM is performed on a local context (*i.e.*, the output of Cross-modal Transformer), while MFM is performed on a global context (*i.e.*, the output of Temporal Transformer). The model is trained to reconstruct masked frames $\mathbf{v}_{\mathbf{m}}$, given the remaining frames $\mathbf{v}_{\backslash \mathbf{m}}$ and all the subtitle sentences $\mathbf{s}$. The visual features of masked

[4] $\mathbb{N}$ is a natural number, M is the number of masked tokens, and $\mathbf{m}$ is the set of masked indices.

[5] Following BERT, we decompose the 15% randomly masked-out words into 10% random words, 10% unchanged, and 80% `[MASK]`.

frames are replaced by zeros. Unlike textual tokens that are represented as discrete labels, visual features are high-dimensional and continuous, thus cannot be supervised via class likelihood. Instead, we propose two variants for MFM, which share the same objective base:

$$\mathcal{L}_{\text{MFM}}(\theta) = \mathbb{E}_D f_\theta(\mathbf{v_m}|\mathbf{v}_{\backslash\mathbf{m}}, \mathbf{s}) \,. \quad (4)$$

Masked Frame Feature Regression (MFFR) MFFR learns to regress the output on each masked frame $\mathbf{v}_\mathbf{m}^{(i)}$ to its visual features. Specifically, we apply an FC layer to convert the output frame representations into a vector $h_\theta(\mathbf{v}_\mathbf{m}^{(i)})$ of the same dimension as the input visual feature $r(\mathbf{v}_\mathbf{m}^{(i)})$. Then we apply L2 regression between the two: $f_\theta(\mathbf{v_m}|\mathbf{v}_{\backslash\mathbf{m}}, \mathbf{s}) = \sum_{i=1}^{M} \|h_\theta(\mathbf{v}_\mathbf{m}^{(i)}) - r(\mathbf{v}_\mathbf{m}^{(i)})\|_2^2$.

Masked Frame Modeling with Noise Contrastive Estimation (MNCE) Instead of directly regressing the real values of masked visual features, we use the softmax version of Noise Contrastive Estimation (NCE) loss (Jozefowicz et al., 2016), which is widely adopted in self-supervised representation learning (Sun et al., 2019a; Hjelm et al., 2019; Oord et al., 2018). NCE loss encourages the model to identify the correct frame (given the context) compared to a set of negative distractors.

Similar to MFFR, we feed the output of the masked frames $\mathbf{v}_\mathbf{m}^{(i)}$ into an FC layer to project them into a vector $g_\theta(\mathbf{v}_\mathbf{m}^{(i)})$. Moreover, we randomly sample frames from the output of unmasked frames as negative distractors $\mathbf{v_{neg}} = \{\mathbf{v}_\mathbf{neg}^{(j)} | \mathbf{v}_\mathbf{neg}^{(j)} \in \mathbf{v}_{\backslash\mathbf{m}}\}$, which are also transformed through the same FC layer as $g_\theta(\mathbf{v}_\mathbf{neg}^{(j)})$. The final objective minimizes the NCE loss: $f_\theta(\mathbf{v_m}|\mathbf{v}_{\backslash\mathbf{m}}, \mathbf{s}) = \sum_{i=1}^{M} \log \text{NCE}(g_\theta(\mathbf{v}_\mathbf{m}^{(i)}) | g_\theta(\mathbf{v_{neg}}))$.

3.2.3 Video-Subtitle Matching

The inputs to VSM are: (i) a sampled query s_q from all subtitle sentences; (ii) the whole video clip $\mathbf{v}$; and (iii) the remaining subtitle sentences $\mathbf{s}_{\backslash q}$ for the video clip. We expect the model to learn: (i) *local alignment* - the start and end index $y_{st}, y_{ed} \in \{1, ..., N_v\}$, indicating the span of visual frames aligned with the query;[6] and (ii) *global alignment* - to which video the sampled query is matched.

[6] Timestamps are used to perform local alignment, which are either included with video (e.g., TV) or generated by ASR (e.g., HowTo100M). Refer to Appendix A.4 for details.

In VSM, we follow XML (Lei et al., 2020b) to compute the matching scores between the query and visual frames at both local and global levels. Specifically, we extract the output of Temporal Transformer as the final visual frame representation $\mathbf{V}^{temp} \in \mathbb{R}^{N_v \times d}$. The query is fed into Cross-modal Transformer to compute its textual representations $\mathbf{W}_{s_q}^{cross} = f_{cross}(\mathbf{0}, \mathbf{W}_{s_q}^{embed})$. Based on this, we use a query encoder (Lei et al., 2020b), consisting of a self-attention layer, two linear layers and an LN layer, to obtain the final query vector $\mathbf{q} \in \mathbb{R}^d$ from $\mathbf{W}_{s_q}^{cross}$.

Local Alignment The local query-video matching score is computed using dot product:

$$S_{local}(s_q, \mathbf{v}) = \mathbf{V}^{temp}\mathbf{q} \in \mathbb{R}^{N_v} \,. \quad (5)$$

Two trainable 1D convolution filters are applied to the scores, followed by a softmax layer, to generate two probability vectors $\mathbf{p}_{st}, \mathbf{p}_{ed} \in \mathbb{R}^{N_v}$, representing the probabilities of every position being the start and end of the ground-truth span. During training, we sample 15% subtitle sentences as queries for each video, and use the cross-entropy loss to predict the start and end index for local alignment:

$$\mathcal{L}_{local} = -\mathbb{E}_D \log(\mathbf{p}_{st}[y_{st}]) + \log(\mathbf{p}_{ed}[y_{ed}]) \,,$$

where $\mathbf{p}[y]$ denotes indexing the y-th element of the vector $\mathbf{p}$.

Note that, XML computes the query-video matching score for each modality separately, and the final matching score is the sum of the two scores. In our HERO model, multimodal fusion is performed in a much earlier stage.

Global Alignment The global matching score is computed by max-pooling the cosine similarities between each frame and the query:

$$S_{global}(s_q, \mathbf{v}) = \max\left(\frac{\mathbf{V}^{temp}}{||\mathbf{V}^{temp}||}\frac{\mathbf{q}}{||\mathbf{q}||}\right) \,. \quad (6)$$

We use a combined hinge loss $\mathcal{L}_h$ (Yu et al., 2018a) over positive and negative query-video pairs. For each positive pair $(s_q, \mathbf{v})$, we replace $\mathbf{v}$ or s_q with one other sample from in the same mini-batch to construct two sets of negative examples: $(s_q, \hat{\mathbf{v}})$ and $(\hat{s}_q, \mathbf{v})$. The training loss is specified as:

$$\begin{aligned}
&\mathcal{L}_h(S_{pos}, S_{neg}) = \max(0, \delta + S_{neg} - S_{pos}) \,, \\
&\mathcal{L}_{global} = -\mathbb{E}_D[\mathcal{L}_h(S_{global}(s_q, \mathbf{v}), S_{global}(\hat{s}_q, \mathbf{v})) \\
&+ \mathcal{L}_h(S_{global}(s_q, \mathbf{v}), S_{global}(s_q, \hat{\mathbf{v}}))] \,, \quad (7)
\end{aligned}$$

where δ is the margin hyper-parameter. The final loss $\mathcal{L}_{\text{VSM}} = \lambda_1 \mathcal{L}_{local} + \lambda_2 \mathcal{L}_{global}$, where λ_1 and λ_2 are hyper-parameters balancing the two terms.

3.2.4 Frame Order Modeling

The inputs for FOM are: (i) all subtitle sentences $\mathbf{s}$; (ii) visual frames $\mathbf{v}$; and (iii) the reorder indices $\mathbf{r} = \{r_i\}_{i=1}^{R} \in \mathbb{N}^R$.[7] We randomly select 15% of the frames to be shuffled, and the goal is to reconstruct their original timestamps, denoted as $\mathbf{t} = \{t_i\}_{i=1}^{R}$, where $t_i \in \{1, ..., N_v\}$. We formulate FOM as a classification problem, where $\mathbf{t}$ is the ground-truth labels of the reordered frames.

Specifically, reordering happens after the multimodal fusion of subtitle and visual frames. The reordered features are fed into Temporal Transformer to produce reordered visual frame embeddings $\mathbf{V}_{\mathbf{r}}^{temp}$. These embeddings are transformed through an FC layer, followed by a softmax layer to produce a probability matrix $\mathbf{P} \in \mathbb{R}^{N_v \times N_v}$, where each column $\mathbf{p}_i \in \mathbb{R}^{N_v}$ represents the scores of N_v timestamp classes that the i-th timestamp belongs to. The final objective is to minimize the the negative log-likelihood (cross-entropy loss):

$$\mathcal{L}_{\text{FOM}} = -\mathbb{E}_D \sum_{i=1}^{R} \log \mathbf{P}[r_i, t_i] \,. \quad (8)$$

4 Experiments

In this section, we describe comprehensive experiments on downstream tasks and provide ablation studies for in-depth analysis of different pre-training settings.

To validate the effectiveness of HERO, we evaluate on a wide variety of downstream tasks, including Text-based Video/ Video-moment Retrieval, Video Question Answering, Video-and-language Inference, and Video Captioning. We consider 6 existing benchmarks: TVR (Lei et al., 2020b), TVQA (Lei et al., 2018), VIOLIN (Liu et al., 2020), TVC (Lei et al., 2020b), DiDeMo (Anne Hendricks et al., 2017a), and MSR-VTT (Xu et al., 2016b). Detailed descriptions and evaluation metrics on each task can be found in Appendix A.6.

4.1 Pre-training Datasets

Our pre-training dataset is composed of 7.6M video clips with their accompanying subtitles from TV and HowTo100M datasets. We exclude all the videos that appear in the downstream tasks to avoid contamination in evaluation.

[7] R is the number of reordered frames, and $\mathbf{r}$ is the set of reorder indices.

TV Dataset (Lei et al., 2018) was built on 6 popular TV shows across 3 genres: medical dramas, sitcoms and crime shows. It contains 21,793 video clips from 925 episodes. Each video clip is 60-90 seconds long, covering long-range scenes with complex character interactions and social/professional activities. Dialogue for each video clip is also provided.

HowTo100M Dataset (Miech et al., 2019) was collected from YouTube, mostly instructional videos. It contains 1.22 million videos, with activities falling into 12 categories (e.g., Food & Entertaining, Home & Garden, Hobbies & Crafts). Each video is associated with a narration as subtitles that are either written manually or from an Automatic Speech Recognition (ASR) system. The average duration of videos in HowTo100M is 6.5 minutes. We cut the videos into 60-second clips to make them consistent with the TV dataset, and exclude videos in non-English languages. These pre-processing steps result in a subset of 7.56M video clips, accompanied with English subtitles.

4.2 New Benchmarks

Existing benchmarks are mostly built on videos from either a single domain or a single modality. In order to evaluate on diverse video content that reflects multimodality challenges, we introduce two new datasets as additional benchmarks: *How2R* for text-based video-moment retrieval, and *How2QA* for video question answering.

How2R Amazon Mechanical Turk (AMT) is used to collect annotations on HowTo100M videos. Figure 6a in Appendix shows the interface for annotation. We randomly sample 30k 60-second clips from 9,421 videos and present each clip to the turkers, who are asked to select a video segment containing a single, self-contained scene. After this segment selection step, another group of workers are asked to write descriptions for each displayed segment. Narrations are not provided to the workers to ensure that their written queries are based on visual content only. These final video segments are 10-20 seconds long on average, and the length of queries ranges from 8 to 20 words.

From this process, we have collected 51,390 queries for 24k 60-second clips from 9,371 videos in HowTo100M, on average 2-3 queries per clip. We split the video clips and its associated queries into 80% train, 10% val and 10% test.

Pre-training Data		Pre-training Tasks	TVR			TVQA	How2R			How2QA
			R@1	R@10	R@100	Acc.	R@1	R@10	R@100	Acc.
TV	1	MLM	2.92	10.66	17.52	71.25	2.06	9.08	14.45	69.79
	2	MLM + MNCE	3.13	10.92	17.52	71.99	2.15	9.27	14.98	70.13
	3	MLM + MNCE + FOM	3.09	10.27	17.43	72.54	2.36	9.85	15.97	70.85
	4	MLM + MNCE + FOM + VSM	4.44	14.69	22.82	72.75	2.78	10.41	18.77	71.36
	5	MLM + MNCE + FOM + VSM + MFFR	4.44	14.29	22.37	72.75	2.73	10.12	18.05	71.36
Howto100M	6	MLM + MNCE + FOM + VSM	3.81	13.23	21.63	73.34	3.54	**12.90**	20.85	73.68
TV + HowTo100M	7	MLM + MNCE + FOM + VSM	**5.13**	**16.26**	**24.55**	**74.80**	**3.85**	12.73	**21.06**	**73.81**

Table 1: Evaluation on pre-training tasks and datasets. Dark and light grey colors highlight the top and second best results across all the tasks trained with TV Dataset. The best results are in bold.

How2QA To collect another dataset for video QA task, we present the same set of selected video clips to another group of AMT workers for multi-choice QA annotation. Each worker is assigned with one video segment and asked to write one question with four answer candidates (one correct and three distractors). Similarly, narrations are hidden from the workers to ensure the collected QA pairs are not biased by subtitles.

We observe that human-written negative answers suffer from serious bias (*i.e.*, models can learn to predict correctly without absorbing any information from the video or subtitles). To mitigate this, we use adversarial matching (Zellers et al., 2019) to replace one of the three written negative answers by a correct answer from another question that is most relevant to the current one. Similar to TVQA, we also provide the start and end points for the relevant moment for each question. After filtering low-quality annotations, the final dataset contains 44,007 QA pairs for 22k 60-second clips selected from 9035 videos. We split the data into 80% train, 10% val and 10% test sets. More details about data collection can be found in Appendix A.9.

4.3 Ablation Study

We analyze the effectiveness of model design, especially different combinations of pre-training tasks and datasets, through extensive ablation studies.

Optimal Setting of Pre-training Tasks To search for the optimal setting of pre-training tasks, we conduct a series of extensive ablation studies to test each setting, using video-moment retrieval and QA downstream tasks as evaluation. Table 1 summarizes ablation results on TVR, TVQA, How2R and How2QA under different pre-training settings. Models are trained on TV dataset only for computational efficiency. Compared to using MLM only (L1 in Table 1), adding MNCE (L2) shows improvement on all downstream tasks. The best performance is achieved by MLM + MNCE + FOM + VSM (L4).

Effect of FOM and VSM When MLM, MNCE and FOM are jointly trained (L3), there is a large performance gain on TVQA, and significant improvement on How2R and How2QA. Comparable results are achieved on TVR. This indicates that FOM, which models sequential characteristics of video frames, can effectively benefit downstream tasks that rely on temporal reasoning (such as QA tasks).

We observe significant performance lift by adding VSM (L4), and the local and global alignments between subtitle and visual frames learned through VSM are especially effective on TVR and How2R. Adding additional MFFR (L5) reaches slightly worse results. Our observation is that MFFR is competing with (instead of complimentary to) MNCE during pre-training, which renders the effect of MFFR negligible.

Effect of Pre-training Datasets We study the effect of pre-training datasets by comparing TV dataset with HowTo100M. In this study, we first pre-train our model on HowTo100M dataset (L6). We observe a performance drop on TVR, while a performance boost on TVQA, How2R and How2QA, compared to the model trained on TV dataset (L4). Our hypothesis is that text-based video-moment retrieval is more sensitive to video domains. Although HowTo100M dataset contains much more videos, the model still benefits more from being exposed to similar TV videos during pre-training.

Hierarchical Design vs. Flat Architecture To validate the effectiveness of our model design, we compare HERO with two baselines (with and without pre-training): (i) Hierarchical Transformer (H-TRM) baseline, constructed by simply replacing the Cross-modal Transformer with a RoBERTa model

Method \Task	TVR			How2R			TVQA	How2QA	VIOLIN	TVC			
	R@1	R@10	R@100	R@1	R@10	R@100	Acc.	Acc.	Acc.	Bleu	Rouge-L	Meteor	Cider
SOTA Baseline	3.25	13.41	30.52	2.06	8.96	13.27	70.23	-	67.84	10.87	32.81	16.91	45.38
HERO	**6.21**	**19.34**	**36.66**	**3.85**	**12.73**	**21.06**	**73.61**	**73.81**	**68.59**	**12.35**	**34.16**	**17.64**	**49.98**

(a) Results on multi-channel (video+subtitle) tasks: TVR[12], How2R, TVQA, How2QA, VIOLIN and TVC.

Method \Task	DiDeMo			DiDeMo w/ ASR			MSR-VTT			MSR-VTT w/ ASR		
	R@1	R@10	R@100	R@1	R@10	R@100	R@1	R@5	R@10	R@1	R@5	R@10
SOTA Baseline	1.59	6.71	25.44	-	-	-	14.90	40.20	52.80	-	-	-
HERO	**2.14**	**11.43**	**36.09**	**3.01**	**14.87**	**47.26**	**16.80**	**43.40**	**57.70**	**20.50**	**47.60**	**60.90**

(b) Results on DiDeMo and MSR-VTT with video-only inputs (single-channel), compared with ASR-augmented inputs (multi-channel).

Table 3: Results on the test set of six downstream tasks, compared to task-specific state-of-the-art (SOTA) models: XML (Lei et al., 2020b) for TVR, How2R and DiDeMo, HowTo100M (Miech et al., 2019) for MSR-VTT, STAGE (Lei et al., 2020a) for TVQA (inapplicable to How2QA due to region-level features), Multi-stream (Liu et al., 2020) for VIOLIN, and MMT (Lei et al., 2020b) for TVC.

and encoding subtitles only;[8] (*ii*) Flat BERT-like encoder (F-TRM).[9]

For this ablation experiment, we use TVR and TVQA as evaluation tasks. Results are summarized in Table 2: (*i*) Without pre-training, F-TRM is much worse than HERO on both tasks. This is due to H-TRM and HERO's explicit exploitation of the temporal alignment between two modalities of videos. (*ii*) Pre-training lifts HERO performance by a large margin, but not much for F-TRM or H-TRM. This indicates that cross-modal interactions and temporal alignments learned by HERO through pre-training can provide better representations for downstream tasks.

HERO vs. SOTA with and w/o Pre-training We compare HERO with task-specifc state of the art (SOTA) models, including XML (Lei et al., 2020b) for TVR and STAGE (Lei et al., 2020a) for TVQA. As shown in Table 2, our model consistently outperforms SOTA models on both tasks, with or without pre-training. Note that for TVQA, STAGE is trained with additional supervision on spatial grounding with region-level features for each frame. Without additional supervisions, HERO is able to achieve better performance.

[8]The inputs to Temporal Transformer in H-TRM are the summation of initial frame embedding and max-pooled subtitle embeddings from RoBERTa.

[9]F-TRM takes as input a single sequence by concatenating the embeddings of visual frames and all subtitle sentences, and encodes them through one multi-layer Transformer.

[10]Model parameters are initialized with RoBERTa weights following Lei et al. (2020b).

[11]F-TRM is pre-trained with MLM+MNCE. VSM and FOM cannot be directly applied.

Pre-training	Model	TVR			TVQA
		R@1	R@10	R@100	Acc.
No[10]	SOTA	2.76	9.08	15.97	70.50
	F-TRM	1.99	7.76	13.26	31.80
	H-TRM	2.97	10.65	18.68	70.09
	HERO	2.98	10.65	18.25	70.65
Yes	F-TRM[11]	2.69	9.21	15.98	49.12
	H-TRM	3.12	11.08	18.42	70.03
	HERO	**4.44**	**14.69**	**22.82**	**72.75**

Table 2: Ablation study on model design, comparing HERO to a flat BERT-like encoder (F-TRM) baseline, a Hierarchical Transformer (H-TRM) baseline, and task-specific SOTA models on TVR and TVQA val set.

Key Conclusions The main observations from these extensive ablation studies are summarized as follows:

- The optimal pre-training setting is MLM + MNCE + FOM + VSM, when trained on HowTo100M dataset and TV dataset.
- FOM effectively helps downstream tasks that rely on temporal reasoning (e.g., video QA tasks).
- VSM encourages frame-subtitle alignment, which is especially effective for video-moment retrieval tasks.
- The hierarchical design in HERO explicitly aligns subtitles and frames, while a flat model architecture can only learn this alignment through implicit attention.
- HERO consistently outperforms SOTA with and without pre-training, which further demonstrates the effectiveness of HERO model design.

4.4 Results on Downstream Tasks

Table 3 reports HERO results on the test splits of all downstream tasks. HERO is pre-trained on both TV and HowTo100M datasets, with the optimal pre-training setting: MLM + MNCE + FOM + VSM. We compare HERO with task-specific SOTA models on each downstream task, including: XML (Lei et al., 2020b) for TVR, Didemo and How2R; HowTo100M (Miech et al., 2019) for MSR-VTT; STAGE (Lei et al., 2020a) for TVQA; Multi-stream (Liu et al., 2020) for VIOLIN; and MMT (Lei et al., 2020b) for TVC. Note that we cannot directly apply STAGE to How2QA, as it was specifically designed to leverage region-level features. Our HERO model achieves new state of the art across all benchmarks.

Results on Multi-channel Tasks Table 3a shows results on downstream tasks consisting of multi-channel videos (video + subtitle). On TVR R@1, HERO results nearly double those from XML.[12] Further, without leveraging fine-grained region-level features, HERO outperforms baseline models by +3.28% on TVQA and +0.75% on VIOLIN. When evaluated on TVC, video and subtitles are encoded by HERO, then fed into a 2-layer Transformer decoder to generate captions. Even though no pre-training was applied to the decoder, HERO surpasses SOTA baseline across all metrics, especially +4.60% on Cider. In addition, HERO establishes a strong baseline for new benchmarks How2R and How2QA.

Results on Single-channel Tasks Table 3b presents results on DiDeMo for text-based video-moment retrieval task and MSR-VTT for text-based video retrieval task. On DiDeMo, HERO surpasses XML by +0.55/+4.72/+10.65 on R@1/10/100, without leveraging Temporal Endpoint Feature used in XML. On MSRVTT, HERO outperforms existing video pre-training model (HowTo100M) by +1.9/+3.2/+4.9 on R@1/5/10.

To evaluate in multi-channel setting, we also fine-tuned HERO on MSR-VTT and DiDeMo using both video channel and extracted subtitle channel (with ASR tools). When augmenting DiDeMo/MSR-VTT with ASR inputs, HERO performance is further improved. Although our model design focuses on truly multimodal videos (video+subtitle input), these results demonstrate HEROs superior generalizability to different video types (multi- and single-channel). More results and analysis are provided in Appendix A.1.

[12] To be consistent with TVR leaderboard, results are reported on tIoU>0.7 without nms.

5 Conclusion

In this paper, we present a hierarchical encoder for video+language omni-representation pre-training. Our HERO model presents a hierarchical architecture, consisting of Cross-modal Transformer and Temporal Transformer for multi-modal fusion. Novel pre-training tasks are proposed to capture temporal alignment both locally and globally. Pre-trained on two large-scale video datasets, HERO exceeds state of the art by a significant margin when transferred to multiple video-and-language tasks. Two new datasets on text-based video-moment retrieval and video QA are introduced to serve as additional benchmarks for downstream evaluation. We consider extension of our model to other video-and-language tasks as future work, as well as developing more well-designed pre-training tasks.

References

Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. 2019. Fusion of detected objects in text for visual question answering. In *EMNLP*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017a. Localizing moments in video with natural language. In *ICCV*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017b. Localizing moments in video with natural language. In *CVPR*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. 2020a. Distilling the knowledge of bert for text generation. In *ACL*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020b. Uniter: Universal image-text representation learning. In *ECCV*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NeurIPS*.

Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. 2019. Temporal localization of moments in video collections with natural language. *arXiv preprint arXiv:1907.12763*.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *ICCV*.

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. 2020. Large-scale adversarial training for vision-and-language representation learning. *arXiv preprint arXiv:2006.06195*.

Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. 2017. Semantic compositional networks for visual captioning. In *CVPR*.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *CVPR*.

Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *ICLR*.

Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. In *EMNLP*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *ICCV*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. 2018. Tvqa: Localized, compositional video question answering. In *EMNLP*.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020a. Tvqa+: Spatio-temporal grounding for video question answering. In *ACL*.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020b. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*.

Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. 2020a. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020b. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL*.

Jingzhou Liu, Wenhu Chen, Yu Cheng, Zhe Gan, Licheng Yu, Yiming Yang, and Jingjing Liu. 2020. Violin: A large-scale dataset for video-and-language inference. In *CVPR*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *CVPR*.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. 2020. Univilm: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*.

Tegan Maharaj, Nicolas Ballas, Anna Rohrbach, Aaron Courville, and Christopher Pal. 2017. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *CVPR*.

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*.

Vishvak Murahari, Dhruv Batra, Devi Parikh, and Abhishek Das. 2020. Large-scale pretraining for visual dialog: A simple state-of-the-art baseline. In *ECCV*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*.

Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2016. Jointly modeling embedding and translation to bridge video and language. In *CVPR*.

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

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Q-bert: Hessian based ultra low precision quantization of bert. In *AAAI*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*.

Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. 2019a. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019b. Videobert: A joint model for video and language representation learning. In *ICCV*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019c. Patient knowledge distillation for bert model compression. In *EMNLP*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *CVPR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

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

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence-video to text. In *ICCV*.

Alex Wang and Kyunghyun Cho. 2019. Bert has a mouth, and it must speak: Bert as a markov random field language model. *arXiv preprint arXiv:1902.04094*.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Qiaolin Xia, Haoyang Huang, Nan Duan, Dongdong Zhang, Lei Ji, Zhifang Sui, Edward Cui, Taroon Bharti, and Ming Zhou. 2020. Xgpt: Cross-modal generative pre-training for image captioning. *arXiv preprint arXiv:2003.01473*.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*.

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

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016b. Msr-vtt: A large video description dataset for bridging video and language. CVPR.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. 2018a. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*.

Youngjae Yu, Jongseok Kim, and Gunhee Kim. 2018b. A joint sequence fusion model for video question answering and retrieval. In *ECCV*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *CVPR*.

Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. 2019. Grounded video description. In *CVPR*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *AAAI*.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018a. Towards automatic learning of procedures from web instructional videos. In *AAAI*.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018b. End-to-end dense video captioning with masked transformer. In *CVPR*.

Linchao Zhu and Yi Yang. 2020. Actbert: Learning global-local video-text representations. In *CVPR*.

A Appendix

A.1 Additional Experiments

For further analysis, Table 4 provides comparison between HERO and task-specific SOTA models on the validation splits of each downstream task.[13] For fair comparison, we re-run XML (Lei et al., 2020b) and MMT (Lei et al., 2020b) experiments using our visual frame features, which achieve slightly better performance than the reported results in Lei et al. (2020b). Note that we cannot directly apply our frame-level visual features to STAGE (Lei et al., 2020a) and Multi-stream (Liu et al., 2020), which require region-level features for each video frame.

Overall, HERO achieves state-of-the-art results on all downstream tasks. Our model consistently outperforms XML on both TVR and How2R, with or without pre-training. Table 5 also provides detailed results on TVR and How2R in three different evaluation settings from Lei et al. (2020b): (i) Video Retrieval, (ii) Moment Retrieval, and (iii) Video-moment Retrieval. For both TVR and How2R, pre-training significantly lifts model performance in all three settings. Following Chen et al. (2020b); Lu et al. (2019), we assess the embeddings learned in pre-training before any fine-tuning occurs. On How2R, HERO without fine-tuning achieves (2.11, 9.09, 14.83) for (R1, R10, R100). While the performance is significantly lower than the fine-tuned model (-1.62 for R1), it performs reasonably well without seeing any How2R query, indicating that HERO has learned to align videos and subtitles (pseudo-query) during pre-training.

Note that for TVQA, STAGE is trained with additional supervision on spatial grounding, which requires region-level features for each frame of the video. Without additional supervision on spatial grounding or fine-grained region-level features, HERO is able to achieve better performance than STAGE on TVQA dataset. We also observe that pre-training significantly boosts the performance of HERO across TVR, How2R and TVQA tasks.

On How2QA, since STAGE was specifically designed to leverage region-level features, we cannot directly apply STAGE. Thus, we only compare HERO performance w/o and with pre-training. Results exhibit consistent patterns observed on other downstream tasks: pre-training achieves better performance than w/o pre-training.

[13] For VIOLIN, we report results on test set for fair comparison, since no validation results are reported in Liu et al. (2020).

Pre-training greatly lifts HERO performance on VIOLIN by approximately $+2.9\%$. However, HERO, without pre-training, presents worse performance than the SOTA baseline. Unlike Multi-stream, which leverages fine-grained region-level features, our results are reported on global frame-level features. Therefore, it may be difficult for HERO to capture the inconsistency between hypothesis and video content. For example, changes of hypotheses about region-level attributes (color, shape, and etc.) may result in different conclusions. Extending HERO for region-level video representations could be an interesting future direction.

HERO is also extensible to generation task: multi-modal video captioning. Our results on TVC show that HERO with pre-training surpasses MMT by a large margin. Although pre-training is only applied to the encoder, it significantly improves HERO performance on TVC across all metrics. When no pre-training is applied, HERO is slightly inferior to the SOTA baseline. Our hypothesis is that TVC has short video context (with video length of 9-second on average) but our model is designed for long video representation learning (TVR/TVQA with video length of 76-second on average). How to design pre-training tasks for MMT on TVC or including decoder pre-training for HERO are left for future works.

A.2 Qualitative Analysis

Visualization of VSM One way to understand how HERO aligns subtitles with video frames is to visualize the Video-Subtitle Matching pre-training task. We provide some examples of the top-1 moment predictions for VSM on both TV and HowTo100M corpora. As shown in Figure 2, the predicted moments (red) largely overlap with the ground truth moments (green) with minor differences. In Figure 2a, we human could probably identify the moment by the speaker information and the visual clue of character's emotion. For Figure 2b, objects (rubber bands) might be the key matching clue. The success of HERO to correctly match the moments might be a positive signal that its pre-training captures those human-identified patterns, hence leads to its strong video understanding capability. However, more thorough analysis, both quantitative and qualitative, is needed to interpret what video-language pre-trained models have learned, which we leave to future works.

Method \Task	TVR			How2R			TVQA	How2QA	VIOLIN	TVC			
	R@1	R@10	R@100	R@1	R@10	R@100	Acc.	Acc.	Acc.	Bleu	Rouge-L	Meteor	Cider
SOTA baseline	2.62	8.45	14.86	1.97	8.32	13.45	70.50	-	67.84	10.53	32.35	16.61	44.39
SOTA baseline †	2.76	9.08	15.97	2.06	8.96	13.27	-	-	-	10.90	32.68	16.83	45.86
HERO w/o pre-training[10]	2.98	10.65	18.42	2.17	9.38	15.65	70.65	71.36	65.72	10.75	32.72	16.42	43.62
HERO w/ pre-training	**5.13**	**16.26**	**24.55**	**3.85**	**12.73**	**21.06**	**74.80**	**73.81**	**68.59**	**12.25**	**34.10**	**17.54**	**50.46**

Table 4: Results on the validation set of six multi-channel video downstream tasks, compared to task-specific SOTA models: XML (Lei et al., 2020b) for TVR and How2R, STAGE (Lei et al., 2020a) for TVQA (inapplicable to How2QA due to region-level features), Multi-stream (Liu et al., 2020) for VIOLIN, and MMT (Lei et al., 2020b) for TVC. † indicates re-implementation of the model using our visual frame features.

Downstream Task	Pre-training	Video Ret.			Moment Ret.[18]			Video Moment Ret.[18]		
		R@1	R@10	R@100	R@1	R@10	R@100	R@1	R@10	R@100
TVR	No[10]	19.44	52.43	84.94	3.76	9.59	61.77	2.98	10.65	18.25
	Yes	**30.11**	**62.69**	**87.78**	**4.02**	**10.38**	**62.93**	**5.13**	**16.26**	**24.55**
How2R	No[10]	11.15	39.78	59.62	4.94	12.73	67.90	2.21	9.52	15.17
	Yes	**14.73**	**47.69**	**68.37**	**6.48**	**15.69**	**70.38**	**3.78**	**12.96**	**20.75**

Table 5: Detailed results on TVR and How2R val set, including the main-task (Video Moment Retrieval) and two sub-tasks (Video Retrieval and Moment Retrieval).

Attention Pattern Visualization Following Kovaleva et al. (2019) and Chen et al. (2020b), we analyze observable patterns in the attention maps of HERO. Figure 3 provides visualization examples of the attention maps learned by the Cross-modal Transformer. For completeness, we briefly discuss each pattern here:

- *Vertical*: Attention to a specific frame.
- *Diagonal*: Locally-focused attention to the token/frame itself or preceding/following tokens/frames.
- *Vertical + Diagonal*: Mixture of Vertical and Diagonal.
- *Block*: Intra-modality attention, *i.e.*, textual self-attention or visual self-attention.
- *Heterogeneous*: Diverse attentions that cannot be categorized and highly dependent on actual input.
- *Reversed Block*: Cross-modality attention, *i.e.*, text-to-frame and frame-to-text attention.

Note that we observe patterns slightly different from Chen et al. (2020b): *Vertical* patterns (Figure 3a) are usually over a specific frame instead of special tokens (`[CLS]` or `[SEP]`). We leave more sophisticated attention analysis/probing to future works.

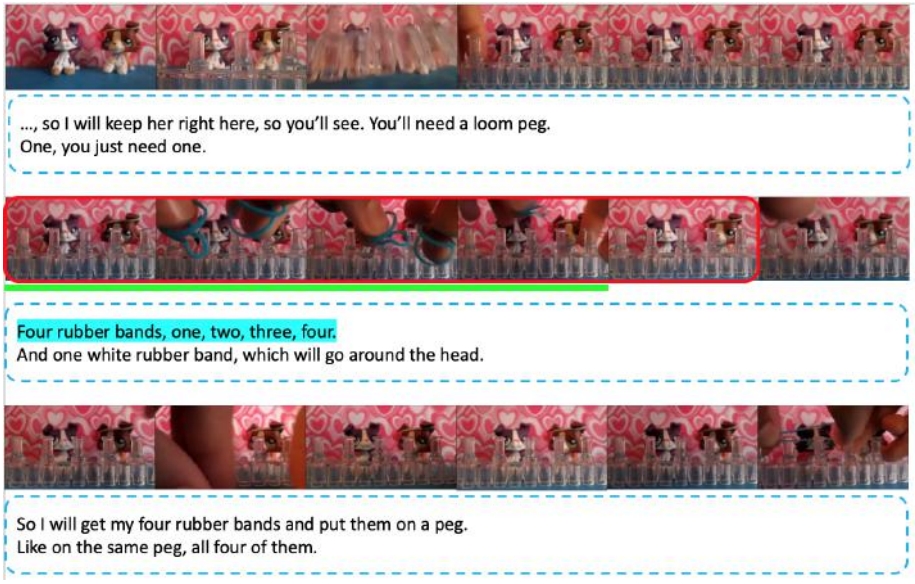
A.3 Downstream Adaptation

The pre-trained model can be readily adapted to downstream video+language tasks through end-to-end finetuning. Below, we describe the detailed adaptation approach to four downstream tasks: (i) text-based video moment retrieval, (ii) video question answering, (iii) video-and-language inference and (iv) multimodal video captioning.

Text-based Video-moment Retrieval The input video clip with accompanying subtitles is encoded by HERO as illustrated in Figure 4. The input query is encoded by the query encoder from the VSM pre-training task. We follow the same procedure as in VSM to compute query-video matching scores both locally (frame-level, for moment retrieval) and globally (clip-level, for video retrieval). The model is finetuned end-to-end using loss $\mathcal{L}_{\text{VSM}}$. Similarly, we let the margin $\delta = 0.1$ and set $\lambda_1 = 0.01$ and $\lambda_2 = 8$ in the loss term $\mathcal{L}_{\text{VSM}}$.

Video Question Answering For Video QA, we consider the multiple-choice setting. As illustrated in Figure 5, for each answer candidate, the corresponding QA pair is appended to each of the subtitle sentences and fed into the Cross-modal Transformer to perform early fusion with local textual context. In addition, these QA pairs are also appended to the input of Temporal Transformer to be fused with global video context. We use a simple attention layer to compute the weighted-sum-across-time of the QA-aware frame representations

(a) TV Dataset.



(b) HowTo100M Dataset.

Figure 2: Visualization of top-1 moment predictions by HERO model for Video-Subtitle Matching on: (a) TV Dataset; and (b) HowTo100M Dataset. Text inside the dashed boxes is the accompany subtitles, with sampled subtitle query highlighted in blue. Groundtruth is highlighted with the green bar under the video frames. Predicted moments are bounded with boxes in red. Best viewed in color.

from the Temporal Transformer output.

These final QA-aware global representations are then fed through an MLP and softmax layer to obtain the probability score $\mathbf{p}_{ans}^{(i)}$ of all the answers for question i. The training objective is

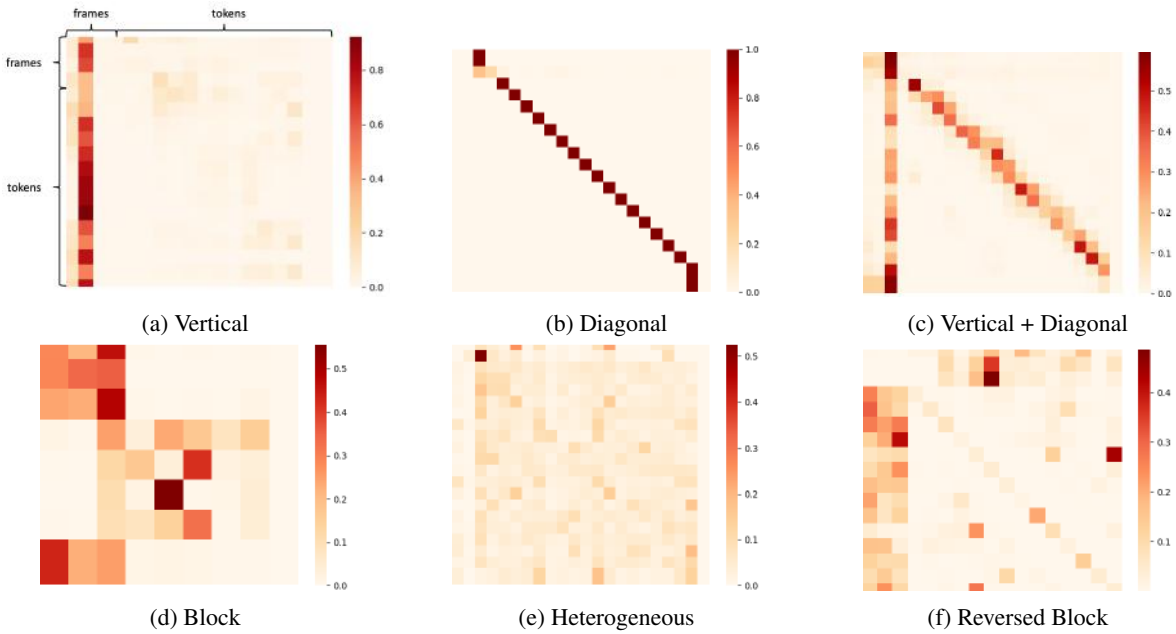
$$\mathcal{L}_{ans} = -\frac{1}{N}\sum_{i=1}^{N} \log \mathbf{p}_{ans}^{(i)}[y_i]\,, \tag{9}$$

where y_i is the index of the ground-truth answer for question i. When supervision is available,[14] we also include the span prediction loss:

$$\mathcal{L}_{span} = -\frac{1}{2N}\sum_{i=1}^{N} (\log \mathbf{p}_{st}^{(i)}[y_i^{st}] + \log \mathbf{p}_{ed}^{(i)}[y_i^{ed}])\,, \tag{10}$$

where $\mathbf{p}_{st}^{(i)}$ and $\mathbf{p}_{ed}^{(i)}$ are the prediction scores of the start and end position, obtained by applying weighted-sum-across-answers attention to the Temporal Transformer output followed by two MLPs and a softmax layer. y_i^{st}, y_i^{ed} are the indices of the

[14] Some existing Video QA tasks require localizing 'frames of interest' for the question, *e.g.*, TVQA+ (Lei et al., 2020a).



(a) Vertical (b) Diagonal (c) Vertical + Diagonal

(d) Block (e) Heterogeneous (f) Reversed Block

Figure 3: Visualization of the attention maps learned by Cross-modal Transformers of HERO model.

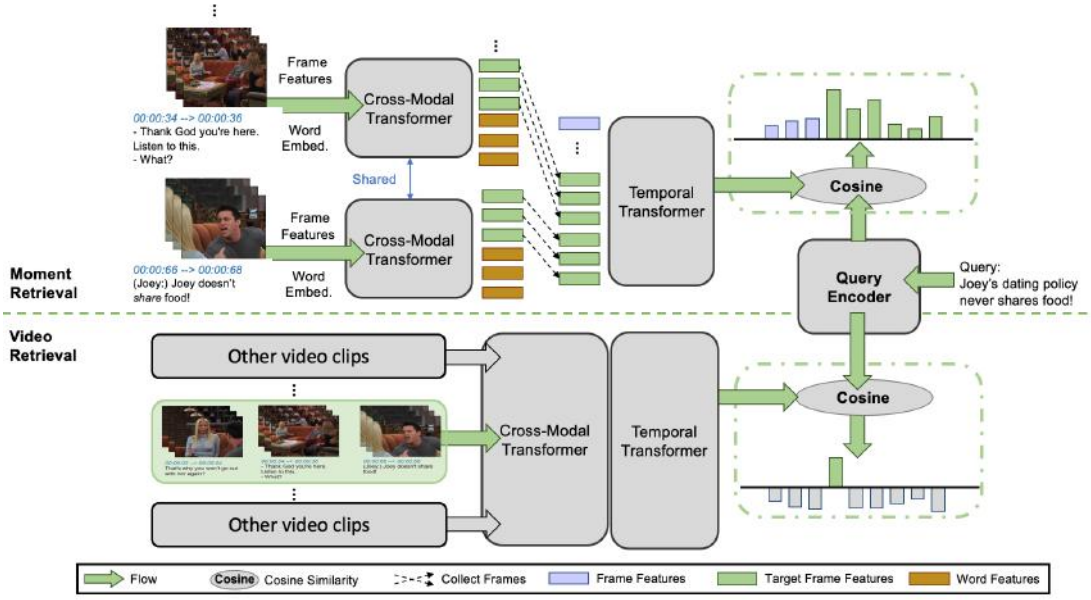


Figure 4: HERO model adapted to downstream task: Text-based Video Moment Retrieval.

ground-truth start and end positions for question i.

The final loss $\mathcal{L}_{\text{QA}} = \mathcal{L}_{ans} + \lambda\mathcal{L}_{span}$, where λ is the hyper-parameter that balance the above two terms. Empirically, we found that $\lambda = 0.5$ yields the best model performance.

Video-and-Language Inference Similar to Video QA, each natural language hypothesis (or query) is appended to each of the subtitle sentences and also to the input of Temporal Transformer. A simple attention pooling layer is added to HERO to obtain the final query-aware global representations.

Video-and-language inference task can be regarded as a binary classification problem. We supervise the training using cross-entropy loss.

Multimodal Video Captioning With a simple addition of a Transformer decoder (Vaswani et al., 2017), we can extend HERO for multimodal video captioning. We feed the whole subtitle-aligned video clip into HERO and obtain the subtitle-fused video representation for each frame. Next, frame representations are grouped by the "moment of interest" using the time interval provided in the cap-

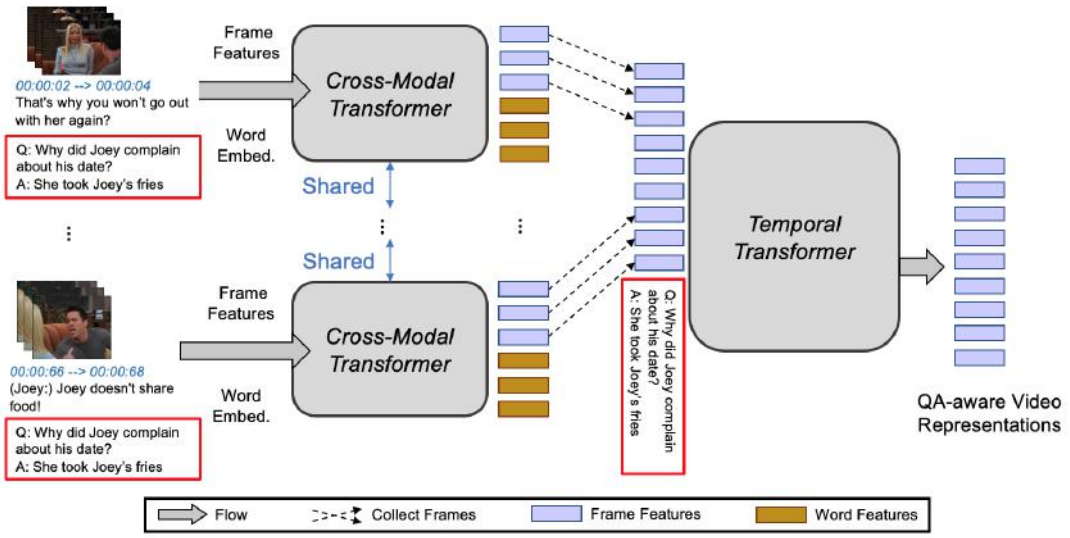


Figure 5: HERO model adapted to downstream task: Video Question Answering.

tion annotation. The decoder-to-encoder attention is applied on the representations of the corresponding video moment and the decoder is trained with conventional left-to-right language modeling cross-entropy loss together with the HERO encoder end-to-end. To make the comparison to MMT (Lei et al., 2020b) as fair as possible, we use shallow Transformer decoder (2-layer) with 768 hidden size. We do not use self-critical RL or its variants to optimize test metrics. Following MMT, greedy decoding is used at inference.

Single-channel Tasks Although HERO is designed for multi-channel videos (video+subtitle), we can easily extend it to single-channel video (video-only) tasks by adding an empty-string subtitle input and pair it with the whole frame sequence. For DiDeMo, we follow the same procedure as in VSM to compute both frame-level (for moment retrieval) and clip-level (for video retrieval) query-video matching scores. For MSR-VTT, a text-based video retrieval task, only clip-level scores are computed.

A.4 Frames/Subtitles Pre-processing

Given a pair of video clip and its associated subtitle, we first extract a sequence of visual frames $\mathbf{v} = \{v_i\}_{i=1}^{N_v}$ at a fixed frame rate (N_v is the number of visual frames in a video clip). The subtitle is parsed into sentences $\mathbf{s} = \{s_i\}_{i=1}^{N_s}$ (N_s is the number of sentences in each subtitle). Note that $N_v \neq N_s$ in most cases, since a subtitle sentence may last for several visual frames. We then align the subtitle sentences temporally with the visual frames. Specifically, for each subtitle sentence s_i, we pair it with a sequence of visual frames whose timestamps overlap with the subtitle timestamp, and denote these visual frames as $\mathbf{v}_{s_i} = \{v_{s_i}^j\}_{j=1}^{K}$ (K is the number of overlapping frames with s_i). In the case that multiple sentences overlap with the same visual frame, we always pair the frame with the one with maximal temporal Intersection over Union (tIoU) to avoid duplication. It is possible that a subtitle sentence is not paired with any visual frame, and in this case, we concatenate it to the neighboring sentences to avoid information loss.

A.5 Implementation Details

We extract 2304-dimensional Slowfast (Feichtenhofer et al., 2019) features at a fixed frame rate (TV: 2/3 frame per second, HowTo100M: 1/2 frame per second). and 2048-dimensional ResNet-101 (He et al., 2016) features at doubled frame rate and max-pooled to get a clip-level feature. The final frame features is concatenation of the two features with dimension 4352. The model dimensions are set to (L=6, H=768, A=12) for Cross-Modal Transformer and (L=3, H=768, A=12) for Temporal Transformer, where L is the number of stacked Transformer blocks; H stands for hidden activation dimension and A is the number of attention heads. For pre-training task VSM, we let the margin $\delta = 0.1$ and set $\lambda_1 = 0.01$ and $\lambda_2 = 8$ in the loss term $\mathcal{L}_{\text{VSM}}$.

Our models are implemented based on PyTorch (Paszke et al., 2017).[15] To speed up training,

[15] https://pytorch.org/

we use Nvidia Apex[16] for mixed precision training. Gradient accumulation (Ott et al., 2018) is applied to reduce multi-GPU communication overheads. All pre-training experiments are run on Nvidia V100 GPUs (32GB VRAM; NVLink connection). We use AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $3e-5$ and weight decay of 0.01 to pre-train our model. The best pre-trained model is trained on 16 V100 GPUs for about 3 weeks. Finetuning experiments are implemented on the same hardware or Titan RTX GPUs (24GB VRAM) with AdamW optimizer but different learning rates.

A.6 Downstream Tasks

TVR (Lei et al., 2020b) is the first to introduce text-based video-moment Retrieval task for multi-channel videos (video+subtitle): given a natural language query, a model is required to not only retrieve the most relevant video clip from the video corpus, but also localize the relevant moment in the retrieved video clip. TVR is built upon the TV dataset, split into 80% train, 10% val, 5% test-public and 5% test-private. On average, 5 queries were collected for each video clip. Among them, 74.2% of queries are related to video only, 9.1% to text only, and 16.6% to both video and text.

TVQA (Lei et al., 2018) was introduced along with the TV dataset. Given a video clip and the accompanying subtitles, the goal is to answer a multiple-choice question about the video. Each video clip has 7 questions, with 5 answers per question. The start/end points of relevant moments are provided for each question.[17]

VIOLIN (Liu et al., 2020) is a new Video-and-Language Inference task. Given a video clip with aligned subtitles as premise, a model needs to infer whether a natural language hypothesis is entailed or contradicted by the given video clip. It consists of 95.3K video-hypothesis pairs from 15.9K video clips, split into 80% train, 10% val and 10% test.

TVC (Lei et al., 2020b) is a multimodal Video Captioning dataset extended from TVR, containing 262K descriptions paired with 108K video moments.[17] Note that it differs from traditional video captioning tasks in that models are allowed to utilize subtitle texts as input.

DiDeMo (Anne Hendricks et al., 2017a) is designed for text-based video-moment retrieval on single-channel videos (video-only). It consists of 10.6K unedited video from Flickr with 41.2K sentences aligned to unique moments in the video. The dataset is split into 80% train, 10% val and 10% test. Note that moment start and end points are aligned to five-second intervals and the maximum annotated video length is 30 seconds.

MSR-VTT (Xu et al., 2016b), for text-based video retrieval on single-channel videos (video-only), includes YouTube videos collected from 257 popular video queries from 20 categories (e.g. music, sports, movie, etc.). It contains 200K unique video clip-caption pairs. We follow the same setup in Yu et al. (2018b) to evaluate our model on MSR-VTT.

Evaluation Metrics Text-based Video-moment Retrieval can be decomposed into two sub-tasks: (i) Video Retrieval: retrieve the most relevant video clip described by the query; (ii) Moment Retrieval: localize the correct moment from the most relevant video clip. A model prediction is correct if: (i) its predicted video matches the ground-truth (in Video Retrieval); and (ii) its predicted span has high overlap with the ground-truth (in Moment Retrieval). Average recall at K (R@K) over all queries is used as the evaluation metric for TVR, How2R, Didemo and MSR-VTT. For TVR, How2R and Didemo, temporal Intersection over Union (tIoU) is used to measure the overlap between the predicted span and the ground-truth span.[18]

TVQA and How2QA include 3 sub-tasks: QA on the grounded clip, question-driven moment localization, and QA on the full video clip. We only consider QA on the full video clip, as it is the most challenging setting among the three. Video clips in VIOLIN are constrained to a single, self-contained scene, hence no additional grounding annotation is provided. Accuracy is used to measure model performance on TVQA, How2QA and VIOLIN.

TVC performance is measured by standard captioning metrics, inlcuding BLEU@4 (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), ROUGE-L (Lin, 2004), and CIDEr-D (Vedantam et al., 2015).

[16] https://github.com/NVIDIA/apex

[17] Train, val and test video splits are the same as TVR.

[18] During evaluation, the average recalls are calculated with tIoU>0.7. we apply non-maximal suppression (nms) with threshold 0.5 to TVR and How2R predictions following Lei et al. (2020b).

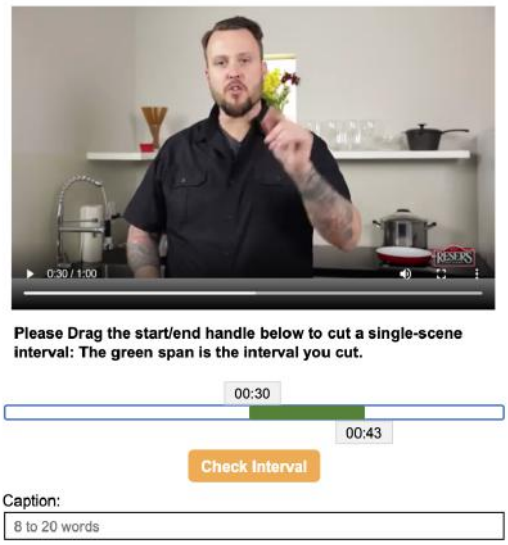
A.7 Vision+Language Pre-training Overview

Very recently, multimodal pre-training has gained increasing attention, especially in the image+text area. Pioneering works such as ViLBERT (Lu et al., 2019) and LXMERT (Tan and Bansal, 2019) propose to encode image and text modalities by two separate Transformers, with a third Transformer for later multimodal fusion. Compared to this two-stream architecture, VL-BERT (Su et al., 2020), Unicoder-VL (Li et al., 2020a), B2T2 (Alberti et al., 2019), VisualBERT (Li et al., 2019), and UNITER (Chen et al., 2020b) advocate single-stream architecture, where image and text signals are fused together in early stage. In VLP (Zhou et al., 2020) and XGPT (Xia et al., 2020), image captioning is considered as additional downstream application, so is visual dialog in Murahari et al. (2020). More recently, ViLBERT is enhanced by multi-task learning (Lu et al., 2020), Oscar (Li et al., 2020b) enhances pre-training with image tags, and Pixel-BERT (Huang et al., 2020) proposes to align image pixels (instead of bottom-up features (Anderson et al., 2018)) with text. Through these pre-training efforts, tremendous progress has been made for vision-and-language representation learning.

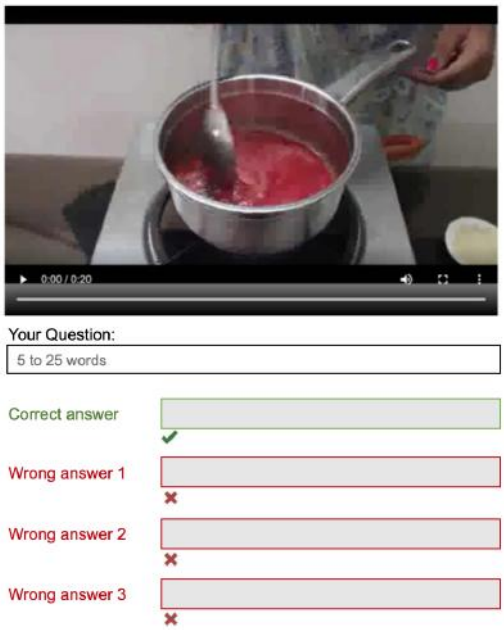
A.8 Video+Language Tasks Overview

Text-based Video-moment retrieval is one of the most popular video+language tasks currently studied. Anne Hendricks et al. (2017b) and Gao et al. (2017) introduce the task of Single Video Moment Retrieval (SVMR), which aims at retrieving a moment from a single video via a natural language query. Escorcia et al. (2019) extends SVMR to Video Corpus Moment Retrieval (VCMR), extending searching pool from single video to large video corpus. TVR (Lei et al., 2020b) defines a new task, Video-Subtitle Corpus Moment Retrieval, which provides temporally aligned subtitle sentences along with the videos as inputs. For this new task, XML (Lei et al., 2020b) is proposed to compute similarity scores between the query and each modality separately (visual frames, subtitles) and then sum them together for final prediction.

Another popular task is Video Question Answering (QA), which aims to predict answers to natural language questions given a video as context. Most previous work focuses on QA pairs from one modality only. For example, MovieFIB (Maharaj et al., 2017) focuses on visual concepts,



(a) User interface for query annotation. Each worker is provided with a video clip and required to select a single-scene clip from the video, then write a query in the text box.



(b) User interface for question/answer annotation. Each worker is provided with a segmented clip and required to write a question with four answers in the text boxes.

Figure 6: Data collection interface: (a) How2R; and (b) How2QA.

MovieQA (Tapaswi et al., 2016) is based on text summaries, and TGIF-QA(Jang et al., 2017) depends on predefined templates for question generation on short GIFs. TVQA (Lei et al., 2018) designed a more realistic multimodal setting: collecting human-written QA pairs along with their associated video segments by providing the an-

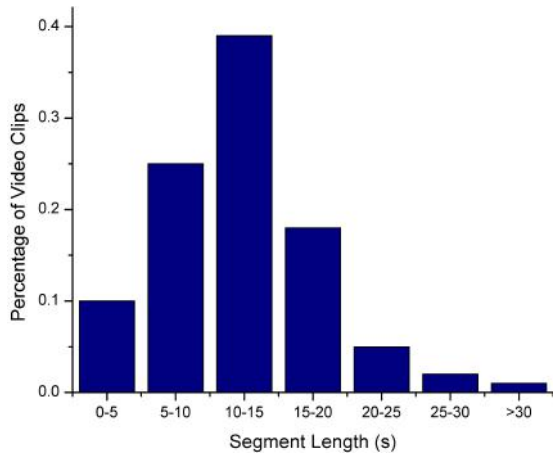


Figure 7: Distribution of video segment length.

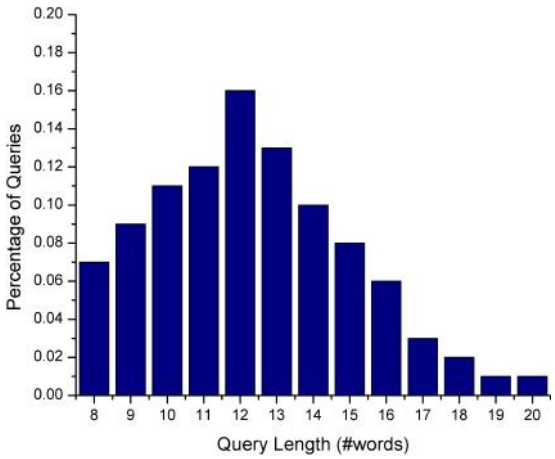


Figure 8: How2R query length distribution.

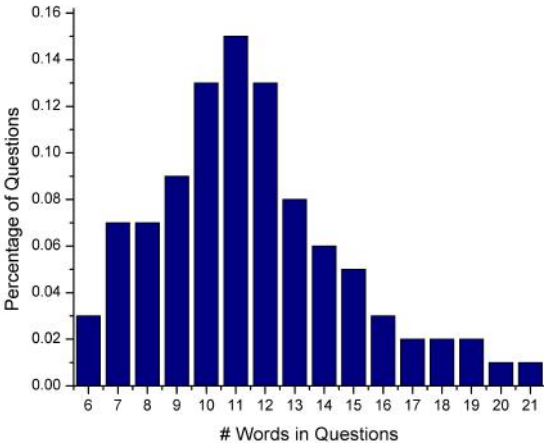


Figure 9: How2QA question length distribution.

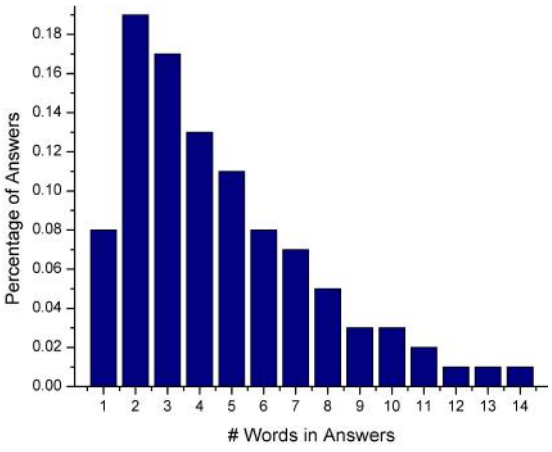


Figure 10: How2QA answer length distribution.

notators with both video clips and accompanying subtitles. Later on, Lei et al. (2020a) augmented TVQA with frame-level bounding box annotations for spatial-temporal video QA, and introduced the STAGE framework to jointly localize moments, ground objects, and answer questions.

Inspired by natural language inference (Bowman et al., 2015; Williams et al., 2018) and visual entailment (Xie et al., 2019), Liu et al. (2020) recently proposed Video-and-Language Inference task along with VIOLIN dataset, which requires a model to draw inference on whether a written statement entails or contradicts a given video clip. This new task is challenging to solve, as a thorough interpretation of both visual and textual clues from videos is required to achieve in-depth understanding and inference for a complex video scenario.

There are also recent studies on video captioning (Venugopalan et al., 2015; Pan et al., 2016; Gan et al., 2017; Zhou et al., 2018b, 2019), popular benchmarks including Youtube2Text (Guadarrama et al., 2013), MSR-VTT (Xu et al., 2016a), YouCook2 (Zhou et al., 2018a), ActivityNet Captions (Krishna et al., 2017) and VATEX (Wang et al., 2019). Unlike previous work mostly focusing on captions describing the visual content, a unique TVC (Lei et al., 2020b) dataset was released with captions that also describe dialogues/subtitles.

A.9 How2R and How2QA Benchmarks

Data Collection Interface Figure 6a and 6b present the interfaces used for collecting How2R and How2QA. For How2R, the annotator is asked to first select a video segment from the presented video clip using the sliding bar, and then enter a description about the selected video segment in the text box (shown at the bottom of Figure 6a). For How2QA, we reuse the selected video segments collected for How2R. The annotators are asked to write a question, a correct answer and 3 wrong answers in the five text boxes shown in Figure 6b.

Video Segment Length Distribution The length

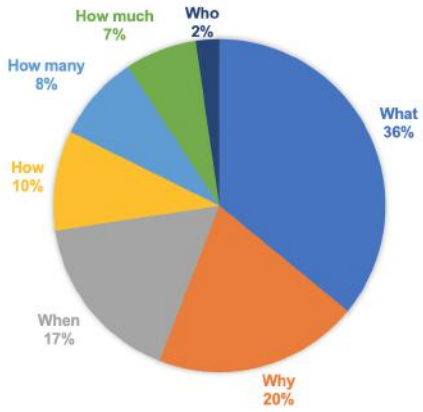


Figure 11: Distribution of questions categorized by their leading words in How2QA.

distribution of selected video segments is presented in Figure 7. The length of video segments varies from 5 to more than 30 seconds. The majority of them have length less than 15 seconds.

How2R Query Length Distribution Figure 8 shows the length (in number of words) distribution of collected queries in How2R. The length of queries is diverse, ranging from 8 to 20.

How2QA Question and Answer Distribution Figure 9 and Figure 10 show the length (in number of words) distribution of collected questions and answers in How2QA. Questions are relatively longer, with more than 10 words on average. Answers are relatively shorter, most of which have less than 7 words.

In addition, we analyze the types of collected question by plotting the distribution of their leading words in Figure 11. In total, we collected questions in 7 different types. Majority of them starts with "what", "why" and "when".