TOKEN-LEVEL CONTRAST FOR VIDEO AND LANGUAGE ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-modal video and language alignment requires grounding linguistic concepts and video to a shared space. Most work neglects explicit token-level grounding, assuming masked token prediction will learn the necessary token-level cross-modal representations. However, it does not force lexical grounding to perception and introduces a domain mismatch between pretraining and fine-tuning. This paper introduces a simple alternative, **To**ken-Level **Co**ntrastive loss (**ToCo**), informed by syntactic classes (*e.g.*, nouns and verbs) to force models to prioritize grounding concrete words. **ToCo** *does not mask inputs* but poses both local (contextual *token*) and global (lexical *type*) pressures for cross-modal alignment in a contrastive manner. Our approach enables a simple vanilla BERT-based multimodal transformer to compete with or outperform existing heavily engineered models on three benchmarks (YouCook2, MSR-VTT and CrossTask). Importantly, **ToCo** is a plug-n-play addition to any architecture, producing consistent improvements across all experimental conditions and visual features.

1 INTRODUCTION

Connecting language and video is key to progress in multimodal research, as it moves beyond grounded objects (language and images) to understanding actions and causality. The grounding of this expanded symbol space (Harnad, 1990), allows us to imbue words with richer notions of meaning (Bisk et al., 2020). However, naïvely combining both modalities does not guarantee the creation of true joint representation, not all words can be grounded in video, and the relative impact of advances in unimodal representations are difficult to disentangle. We introduce a new simple training object **ToCo** which, combined with comprehensive experiments, aims to address these three concerns and provide a solid foundation for future work to build on in this domain.

The availability of large-scale video data (Miech et al., 2019), combined with recent advances in language modeling (Devlin et al., 2019) have made it possible to explore large-scale self-supervised cross-modal learning, with surprising success. Self-supervision requires a loss function that does

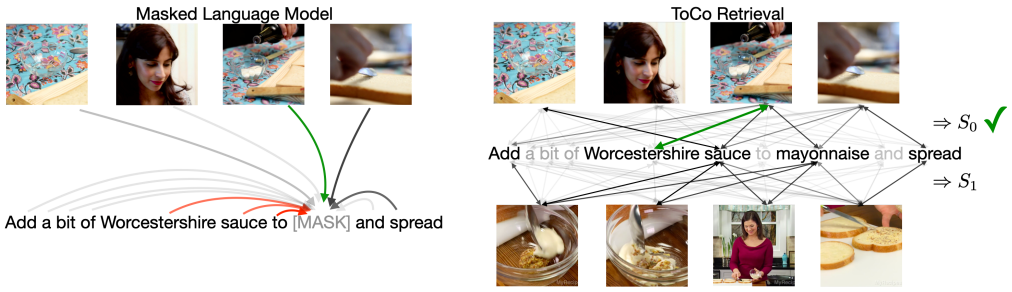


Figure 1: Where a Masked Language Model (left) leads to learning lexical correlations (red) and introduces a domain shift by with the train-only '[MASK]' token, **ToCo** (right) forces the cross-modal fusion to focus on identifying which semantics bearing word differentiates between video sequences. Additionally, note that where a global contrast forces alignment into the model to produce a single representative '[CLS]' token, **ToCo** explicitly guides lexical grounding (dark lines).

not rely on annotations. The Masked-Language-Model (MLM) objective was introduced as a solution for language-only pretraining, and copied over to the cross-modal setting, with the addition of masked features (MFM) (Li et al., 2020a; Zhu & Yang, 2020; Luo et al., 2020). This formulation (1) by design leads to fine-tuning the language model without directly requiring the use of the video signal, (2) it produces a domain mismatch between pretraining and inference where the latter will not have masked inputs. In contrast, we introduce **ToCo** to require type and token-level grounding in every training example, and we do so without the domain mismatch of training time mask tokens.

Fundamental to our approach is token-level contrastive losses which force the grounding of individual words. Given the relative semantic content of nouns, verbs, and adjectives we upweight their contributions to force lexical groundings to fight to rank the correct video highest (given a caption) among a set of NCE examples. In Figure 1, it is clear that the syntactic signals of MLM fight to fill the blank, while in **ToCo** the learning pressure is specifically on *Worcestershire* to differentiate the two videos. This pressure, combined with our merged global and type-token representations (§3.3) push the model to prioritize clean grounded alignments. Note, while we provide a comprehensive analysis on evaluation paradigms and video feature representations, nothing about our approach limits its applicability to video and can be just as readily used in image-text transformers.

In short, this work's contributions are: (1) A cleaner, simpler, and more effective approach to cross-modal learning which can be extended to incorporate linguistic knowledge, (2) A systematic comparison across feature representations, and (3) state-of-the-art results on video benchmarks.

2 Related Work

Effective cross-modal fusion is a key research question for machine learning (Baltrušaitis et al., 2019). Such models enable language interfaces to visual data via retrieval (Lin et al., 2014a; Yu et al., 2016), captioning (Thomason et al., 2014; You et al., 2016; Yang et al., 2016; Wang et al., 2019; Yu et al., 2016), question answering (Antol et al., 2015; Lu et al., 2016; Lei et al., 2018), and reasoning (Zellers et al., 2019). In this paper, we focus on the video and language domain.

Alignment. Joint video and language embeddings enable numerous applications, including retrieval (Lin et al., 2014b; Yu et al., 2017; 2018), captioning (Yu et al., 2016; Zhou et al., 2018), and question answering (Jang et al., 2017; Lei et al., 2018). Compared to image-text alignment (Kiros et al., 2014; Wang et al., 2016; 2018), video requires the model to understand movement and temporal coherence. Some work has relied on attention mechanisms to *extract* key information from videos (Torabi et al., 2016; Yu et al., 2017), while others preserve visual information by composing pairwise joint representation using 3D tensors (Yu et al., 2018) or multi-level video encoders to separately encode the spatial and temporal cues (Dong et al., 2019). These models are usually equipped with a rank or margin loss to learn the correct alignment for video-text pairs.

Pre-training. Large-scale data has been very effective in language representation learning (Devlin et al., 2019), and seamlessly extended to the vision-and-language domain via the addition of bounding box features as visual "words" (Tan & Bansal, 2019; Lu et al., 2019; Zhou et al., 2019; Li et al., 2020b), or video clips as "tokens" (Sun et al., 2019; Zhu & Yang, 2020; Luo et al., 2020; Li et al., 2020a). Pretraining (Miech et al., 2019) significantly improves performance on the aforementioned video-language tasks. Most methods train auxiliary tasks (e.g. video-text alignment, and masked language/frame prediction), but Miech et al. (2020) showed the effectiveness of noise-contrastive estimation (NCE) for learning video and language representations from noisy instructional videos.

3 Method

In this section, we first briefly define the contrastive loss for video and language as background and then introduce our own method. Additional details can be found in the Appendix.

3.1 Preliminary

Given a set of video clips $\mathcal{V} = \{\boldsymbol{v}_1, ..., \boldsymbol{v}_N\}$ and the associated texts $\mathcal{T} = \{\boldsymbol{t}_1, ..., \boldsymbol{t}_N\}$, our goal is to learn a joint embedding such that paired video clips and texts $(\boldsymbol{v}_i, \boldsymbol{t}_i), \forall i \in \{1, ..., N\}$ are well aligned, while distancing from all other pairs $(\boldsymbol{v}_i, \boldsymbol{t}_j), \forall i, j \in \{1, ..., N\}, i \neq j$. This requires both good representations $\boldsymbol{x} = g(\boldsymbol{v})$ and $\boldsymbol{y} = h(\boldsymbol{t})$ of video and text alone, respectively, and a correct joint

alignment function $f(\boldsymbol{x}, \boldsymbol{y})$. In this paper, we assume videos and texts are already well-represented by pre-trained models (*e.g.*, I3D (Carreira & Zisserman, 2017), S3D (Miech et al., 2020) for videos and BERT (Devlin et al., 2019) for text), and merely focus on the learning of $f(\boldsymbol{x}, \boldsymbol{y})$, which outputs a scalar indicating the quality of the alignment for a video-text pair $(\boldsymbol{x}, \boldsymbol{y})$. Below, we denote the features for a video clip by a temporal sequence of m embeddings, $\boldsymbol{x} = \{x^1, ..., x^m\} \in \mathbb{R}^{m \times d_x}$, while a textual narration by a sequence of n embeddings, $\boldsymbol{y} = \{y^1, ..., y^n\} \in \mathbb{R}^{n \times d_y}$.

3.2 Global contrast

A good alignment requires high joint probability $p(\boldsymbol{x}_i, \boldsymbol{y}_j)$ for positive video-text pairs, which should be low for negative pairs. However, it is intractable due to the need to normalize over all video-text pairs, but we can work instead with the conditional $p(\boldsymbol{x}_j|\boldsymbol{y}_i)$ (see Appendix A.1 for details). We require $p(\boldsymbol{x}_i|\boldsymbol{y}_i) > p(\boldsymbol{x}_j|\boldsymbol{y}_i), \forall j \neq i$. This can be achieved by incorporating a margin s.t. $f(\boldsymbol{x}_i, \boldsymbol{y}_i) > f(\boldsymbol{x}_j, \boldsymbol{y}_i) + \delta$ (Li et al., 2020a), optimizing binary cross-entropy s.t. $f(\boldsymbol{x}_i, \boldsymbol{y}_i) = 1$ and $f(\boldsymbol{x}_i, \boldsymbol{y}_j) = 0$, or contrastive loss and its variants (Zhu & Yang, 2020). In this paper, we take the contrastive learning objective. Ideally, our target is $\hat{p}(\boldsymbol{x}_i|\boldsymbol{y}_i) = 1$, while $\hat{p}(\boldsymbol{x}_j|\boldsymbol{y}_i) = 0, \forall j \neq i$. Hence, we can derive the contrastive loss from the cross-entropy:

$$L(\mathcal{X}, \mathcal{Y}) = -\sum_{i=1}^{N} \log(p(\boldsymbol{x}_i|\boldsymbol{y}_i)) = -\sum_{i=1}^{N} \log\left(\frac{\exp^{f(\boldsymbol{x}_i, \boldsymbol{y}_i)}}{\exp^{f(\boldsymbol{x}_i, \boldsymbol{y}_i)} + \sum_{j \neq i} \exp^{f(\boldsymbol{x}_j, \boldsymbol{y}_i)}}\right) \tag{1}$$

$\sum_{k \neq j} \exp^{f(\boldsymbol{x}_k, \boldsymbol{y}_i)}$ is still intractable since it sums over all videos for $\boldsymbol{y}_i$ except for $\boldsymbol{x}_i$ (See Appendix A.1 for derivations). In practice, we can approximate it by sampling a subset of negative video clips from the training data (Miech et al., 2019; Luo et al., 2020; Miech et al., 2020). It has been shown in (Miech et al., 2020) that more negative samples is usually better.

Global alignment function f. Two common approaches for constructing $f(\boldsymbol{x}, \boldsymbol{y})$ are a dot-product of two separate representations, *i.e.*, $f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}\boldsymbol{y}^T$ (Miech et al., 2019; Miech et al., 2020), or via a BERT-like architecture (Li et al., 2020a; Luo et al., 2020; Zhu & Yang, 2020). To capture rich alignment between video and text, we adopt BERT-based architecture in this paper as shown in Fig. 2(a). Given the video representation $\boldsymbol{x} \in \mathcal{R}^{m \times d_x}$ and text representation $\boldsymbol{y} \in \mathcal{R}^{n \times d_y}$, we first embed them separately to the same dimension, *i.e.*, $\hat{\boldsymbol{x}} \in \mathcal{R}^{m \times d}$ and $\hat{\boldsymbol{y}} \in \mathcal{R}^{n \times d}$, and concatenate the sequences to produce $[\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}] \in \mathcal{R}^{(m+n) \times d}$. This combined inputs is passed to a multi-modal encoder consisting of multiple self-attention layers (Vaswani et al., 2017) to produce a contextual video-linguistic representation $[\mathring{\boldsymbol{x}}, \mathring{\boldsymbol{y}}]$. To compute the contrastive loss in equation 1, we take the first entry $\mathring{x}^1$, the '[CLS]' token, and project it into a scalar via a two-layer perceptron (MLP).

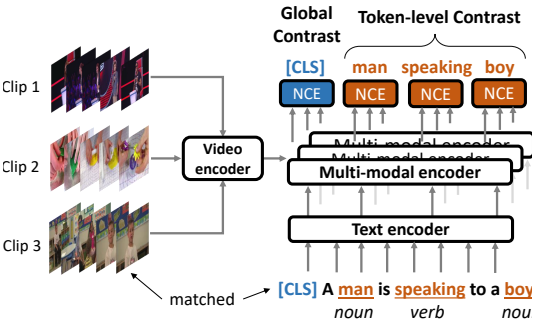
Key to our approach however is that in addition to the use of this "global" representation accessed via the '[CLS]' token we also encourage word-level alignment, via a token-level contrastive method.

3.3 Token-level Contrast

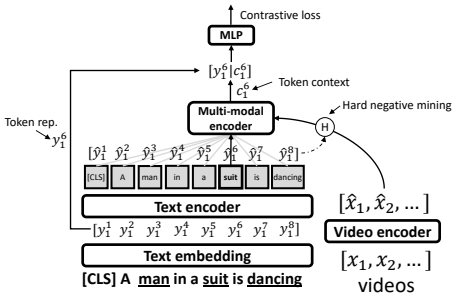
As discussed above, summary representations, while common may not provide the necessary learning pressures to ground individual objects or actions as the pressures are diffuse and allow the model to easily learn spurious correlations between phrases and individual video clips. In order to avoid this, in addition to a global contrastive loss over $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ (discussed above), we apply an objective tying the video clip to the individual tokens. A straightforward way to impose the token-level contrast is composing pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i^k)$ and apply the same contrastive learning discussed above. However, this is prone to fail since a single token is far enough to capture the difference across different videos. To address this, we combine the contextual-independent token representation $\boldsymbol{y}_i^k$ for a single token with the output $\boldsymbol{c}_i^k = \mathring{y}_i^k$ from multi-modal encoder, as shown in Fig. 2(b). Based on this, we can learn a token-level joint alignment function f^t with a contrastive loss:

$$L^t(\mathcal{X}, \mathcal{Y}) = -\sum_{i=1}^{N} \sum_{k \in \mathcal{K}_i} \log\left(\frac{\exp^{f^t(\boldsymbol{x}_i, \boldsymbol{y}_i; y_i^k)}}{\exp^{f^t(\boldsymbol{x}_i, \boldsymbol{y}_i; y_i^k)} + \sum_{j \neq i} \exp^{f^t(\boldsymbol{x}_j, \boldsymbol{y}_i; y_i^k)}}\right) \tag{2}$$

where $\mathcal{K}_i$ are token indices in the i-th text where we want to include the loss. Based on equation 2, the model uses the specific token as an anchor to align with video representation. We refer to this as a **To**ken-Level **Co**ntrastive loss (**ToCo**), because it forces attention to specific tokens, which is complementary to the global counterpart in equation 1.



(a) An illustration of including a token-level contrast. Note NCE losses are not applied to stop-words and are separate from the global [CLS] based loss.



(b) A closeup view of **ToCo** as applied to a single lexical token. Node that the token's unimodal input $\hat{y}_1^6$ is combined with the contextualized representation c_1^6 prior computing the loss.

Figure 2: **ToCo** includes token level losses on semantic bearing words in addition to a global loss (left). The right shows additional details of how representations are shared for a specific token.

Token-level alignment function f^t. To make our model compact, we reuse most part of f to compute f^t. As shown in Fig. 2 (b), for the k-th token, we take the $\mathring{y}_i^k$ output of the cross-modality encoder and concatenate it with the input token representation y_i^k, before passing them to a two-layer perceptron to produce an alignment score. This formulation allows for signal to propagate to both: (1) the global lexical *type* y_i^k for sharing across the data; and (2) $\mathring{y}_i^k$ the contextualized *token* which is output from the cross-modality encoder and contains information from both video clip and text.

Token of interest. Our formulation of a token-level contrastive loss (equation 2) works on separate tokens. This raises the question of which tokens should receive inclusion in **ToCo**. We start heuristically by selecting *nouns*, *verbs* and *adjectives* as the most "groundable" tokens into the video. Next we must determine the relative weight and importance of individual words/classes. A uniform weighting would be equivalent to no-loss, while a very peaked distribution on nouns or verbs might bias towards learning about specific objects or actions. Instead, we compute the TF-IDF (Jones, 1972) weighting for each token to avoid treating all types in a syntactic class the same, as different words appear in different frequencies and thus have different importance even if they are all nouns. The role of linguistic/syntactic prior knowledge (possibly language specific) to ideally weight the relative importance of individual words and phrases is left as a promising exercise for future work.

Hard-negative mining. Previous work has demonstrated that hard negative mining is effective at learning good cross-modality alignment (Lee et al., 2018; Faghri et al., 2017). This strategy can be effectively applied to models with light-weight cross-modality fusion layers, *e.g.*, dot-product layer. However, for a heavy cross-modality encoder such as BERT-like architectures, it is much more time-consuming to compute the alignment scores for all pairs in a mini-batch. To address this issue, we introduce a new hard-negative mining strategy as indicated by the cirled "H" in Fig. 2(b). Assume there is a minibatch of k video-text pairs $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_k\}$ and $\{\boldsymbol{y}_1, ..., \boldsymbol{y}_k\}$, we simply perform average pooling on the frame-wise video features $\boldsymbol{x}_i$ to $\boldsymbol{a}_i \in \mathcal{R}^d$ and the token-wise text features $\boldsymbol{y}_i$ to $\boldsymbol{b}_i \in \mathcal{R}^d$. Then we compute the dot-product between them, resulting in a score matrix of $k \times k$ dimension. During training, we use this score matrix to help the upper multi-modal encoder to select the hard negative samples. At the meaning time, we add another contrastive loss to gradually learn a better matching scoring function. It turns out that this simple strategy works well for both global contrastive loss and our **ToCo** loss.

3.4 Objective Function

The optimal video-linguistic representation minimizes the combination of two contrastive losses:

$$\{f, f^t, h\}^* = \arg\min_{f,f^t,h} \left(L(\mathcal{X}, \mathcal{Y}) + \lambda L^t(\mathcal{X}, \mathcal{Y})\right) \tag{3}$$

where λ is the weight for the token-level contrastive loss. We jointly learn f and f^t, and fine-tune the pre-trained language encoder h as well during the training. During inference, we compute $f(\boldsymbol{x}_i, \boldsymbol{y}_j) + \lambda f^t(\boldsymbol{x}_i, \boldsymbol{y}_j)$ to measure the alignment between $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$.

4 EXPERIMENTS

4.1 DATASETS

In our experiments, we verify its effectiveness on standard benchmark datasets by comparing it with previous work on text-based video retrieval and action localization. We use the three most common datasets and tasks:

- **HowTo100M** (Miech et al., 2019). Howto100M is used for pretraining. It was collected by crawling YouTube, and contains over 1.2M narrated videos associated with automatically generated transcriptions. Each video contains more than 100 video clips on average.
- **YouCook2** (Zhou et al., 2017). YouCook2 is a set of cooking videos which depict routine cooking activities covering 89 recipes and containing 2000 videos. Component video clips are annotated with textual descriptions by human annotators, for a total of 14k (clip, description) pairs.
- **MSR-VTT** (Xu et al., 2016). Compared to YouCook2, MSR-VTT focuses on a more diverse set of activities, but is similar in size with 10K video clips (3K are reserved for testing) Additionally, each clip is associated with 10 human-annotated descriptions. Following Miech et al. (2019), we select 1,000 video clips for the test set to evaluate the video retrieval performance.
- **CrossTask**. We evaluate action localization following the protocol proposed in (Zhukov et al., 2019) and used in (Miech et al., 2019; Miech et al., 2020) to report the average recall.

4.2 VIDEO AND TEXT REPRESENTATIONS

There are three variables at play when learning joint video and language representations: unimodal video, unimodal text, and the method for fusion. As we are introducing a new training regime and loss for fusion, we do our best to compare against every underlying variant of video and language presentations currently used in the literature by Miech et al. (2019); Miech et al. (2020); Zhu & Yang (2020). We summarize these the main settings below:

- **Video Representations**. Miech et al. (2019) use an ImageNet (Russakovsky et al., 2015) pretrained Resnet-152 (He et al., 2015) model to extract a features map at 1fps before pooling to a 2048-dimensional feature vector. For 3D features, a pretrained I3D model (Miech et al., 2019; Miech et al., 2020), R(2+1)D model Zhu & Yang (2020) or S3D model Miech et al. (2020); Luo et al. (2020) are used to extract features from 16 adjacent frames sampled at 24 fps. Likewise, this 3D CNN feature map is then pooled to a 2048-d (I3D) or 1024-d (S3D) feature for each 16 frames. Zhu & Yang (2020) also train an object detection model on Visual Genome (Krishna et al., 2016) to extract objects from the video clips to augment the visual input.
- **Text Representations**. There are primarily two variants of text features: 1) GoogleNews pretrained word2vec (Mikolov et al., 2013) embeddings are used in (Miech et al., 2019; Miech et al., 2020); 2) BERT (Devlin et al., 2019) is used as the backbone for (Zhu & Yang, 2020; Luo et al., 2020). The latter approach is also more common in image-based vision-and-language tasks.

As it is unclear whether the architectures, losses, or features are responsible for the different performances seen on these tasks, to provide fair comparisons to each, we perform separate experiments with each of the the following video representations:

- **I3D-Resnext101**: I3D with Resnext-101 backbone pre-trained on Kinetics-400 (Kay et al., 2017). Miech et al. (2019) and Luo et al. (2020) also concatenated a 2D-Resnet152 feature.
- **I3D-Resnet152**: I3D with Resnet-152 backbone pre-trained on Kinetics-700 (Carreira et al., 2019) used by (Miech et al., 2020) has comparable capacity to R(2+1)D in (Zhu & Yang, 2020).
- **S3D-Howto100M**: S3D pretrained on Howto100M. Miech et al. (2020) demonstrated that S3D significantly outperforms the above features on both video-only and video-text tasks and was integrated into the latest version of (Luo et al., 2020).

We use the off-the-shelf weights of I3D-Resnext101 and I3D-Resnet152 provided by (Hara et al., 2018) and S3D-Howto100M as provided by (Miech et al., 2020), respectively. For all experiments, the maximum number of video and text tokens are set to 48 and 30, respectively. To extract 2D video features, we sample video frames at 1 fps, resulting in one 2048-d 2D-Resnet152 feature per second. For 3D CNN features, we follow Miech et al. (2019) and sample video frames at 24 fps to extract I3D features in a window size of 16, obtaining one and half 2048-d features per second. The textual tokens are first uncased and then fed to a pre-trained BERT-base model. The 768-d outputs of the embedding layer in the BERT-base model are used as the raw token embeddings y_i in equation 2.

Model	Lang.	Video		YouCook2				MSR-VTT			
		2D	3D	R@1	R@5	R@10	Med. R	R@1	R@5	R@10	Med. R
JSFusion (Yu et al., 2018)	BiLSTM	R-152	-	-	-	-	-	10.2	31.2	43.2	13
TVJE (Miech et al., 2019)	w2v	R-152	I-101	**4.2**	**13.7**	**21.5**	**65**	**12.1**	**35.0**	**48.0**	**12**
UniVL-v1 (Luo et al., 2020)	Bert	R-152	I-101	3.4	10.8	17.8	76	14.6	39.0	52.6	10
Our Baseline	Bert	R-152	I-101	3.2	11.1	17.7	81	14.8	41.0	55.8	8
ToCo ($\lambda = 0$)	Bert	R-152	I-101	3.9	12.4	19.1	78	16.8	43.8	57.9	7
ToCo ($\lambda = 0.2$)	Bert	R-152	I-101	**4.6**	13.3	20.0	75	17.4	45.1	58.5	**7**
ToCo ($\lambda = 0.5$)	Bert	R-152	I-101	**4.6**	**14.1**	**20.8**	72	**18.4**	**46.6**	**59.5**	**6**
ToCo ($\lambda = 1.0$)	Bert	R-152	I-101	4.3	13.5	19.9	**71**	18.1	45.7	59.1	**7**
Our Baseline	Bert	R-152	I-152	3.8	13.9	21.3	60	16.5	43.5	56.8	7
ToCo ($\lambda = 0$)	Bert	R-152	I-152	4.1	14.0	21.7	57	16.3	44.7	58.6	7
ToCo ($\lambda = 0.5$)	Bert	R-152	I-152	**4.9**	**15.2**	**22.5**	**55**	**18.0**	**47.3**	**60.3**	**6**
Our Baseline	Bert	-	S-100	12.4	35.6	48.7	11	18.3	45.3	59.6	7
ToCo ($\lambda = 0$)	Bert	-	S-100	14.9	39.3	**52.2**	**9**	19.4	46.3	58.8	6
ToCo ($\lambda = 0.5$)	Bert	-	S-100	**15.9**	**39.7**	51.9	9	**21.1**	**47.9**	**60.5**	**6**

Table 1: Comparing with previous works and baselines under task-specific setting. R-152, I-101 and I-152 are in short of Resnet-152, I3D-Resnext101 and I3D-Resnet152, respectively.

4.3 Settings and Implementation Details

Training on separate datasets. In this setting, we train video retrieval models from scratch (100K iterations) using the training set provided in YouCook2 (Zhou et al., 2017) and MSR-VTT (Xu et al., 2016). We use 64 video-text pairs and sample 8 negative samples either using random sampling or our hard sample mining techniques. We use Adam (Kingma & Ba, 2014) as the optimizer (learning rate of $1e^{-4}$). A linear decay for the learning rate is applied after a warm up of 10k iterations. The weight decay is set to $1e^{-5}$. Training on 4 Nvidia V100 GPUs takes approximately 5 hours.

Pre-training and Finetuning. For pre-training, we make use of all the available videos provided in Howto100M (Miech et al., 2019). We use Adam (Kingma & Ba, 2014) as the optimizer (learning rate of $1e^{-4}$). We train the model for 1M iterations with the same batch size as above, but using 16 negative samples for each pair. We take the final checkpoint and use it for the fine-tuning on separate datasets. We use the same setting as above except for lowering the initial learning rate to $2e^{-5}$.

In the next section, we present experimental results and fair comparisons as possible as wen can to verify the effectiveness of our proposed method.

5 Results

5.1 Task-specific Training

We begin this investigation by exploring the simplest setting of task-specific training only. Here, models do not have access to large-scale pretraining and therefore must use the limited training data effectively. Note, the pretrained unimodal feature representations "bring knowledge" with them, so even in this task-specific setting, like all prior work, the models are not *tabula rasa* learners. Our first result, in Table 1, is that **ToCo** outperforms all existing models across all three video representations.

Weighting ToCo loss. First we explore the importance of our loss by varying the λ of equation 3. We show that the task performance is not simply a function of the transformer architecture, as our baseline result ($\lambda = 0$) performs similarly to previous work, it is the inclusion of the token losses that leads to gains. In the simplest condition (I-101) we do a basic sweep over λ weighting. Note, that while we do see gains from exclusive use of the token loss and no global component ($\lambda = 1$), best performance balances the two. Based on the results in Table 1, we set $\lambda = 0.5$ for all the following experiments. The ideal setting or schedule for this parameter is left to future work.

Video Features The second result to note is the key role video features play in overall performance. Note that improvements in the video features directly correlate to downstream performance, but **ToCo** shows consistent gains regardless of the underlying representation. The most extreme jump is seen in the move to S3D features. This result is present throughout our experiments and so we will continue to provide results on all feature representations to ablate their relative importance.

Figure 3: Qualitative examples of our baseline implementation's rankings of two queries, versus full token-level losses applied (right). Note the down-weighting of videos that are not *stirring* and up-weighting the presence of a *car racing*.

	Model	FT.	#L	Video		YouCook2				MSR-VTT			
				2D	3D	R1	R5	R10	MR	R1	R5	R10	MR
I3D-Resnext101	TJVE (Miech et al., 2019)	✗	n/a	R-152	I-101	6.1	17.3	24.8	46	7.5	21.2	29.6	38
	UniVL-v1 (Luo et al., 2020)	✗	1-2	R-152	I-101	5.5	17.7	27.4	42	2.9	8.3	12.4	173
	ToCo ($\lambda = 0$)	✗	1-2	R-152	I-101	8.3	22.7	31.0	34	3.3	12.3	17.8	87
	ToCo ($\lambda = 0.5$)	✗	1-2	R-152	I-101	**11.1**	**25.9**	**35.4**	**27**	**3.8**	**14.0**	**19.5**	**79**
	TJVE (Miech et al., 2019)	✓	n/a	R-152	I-101	8.2	24.5	35.3	24	14.9	40.2	52.8	9
	UniVL-v1 (Luo et al., 2020)	✓	1-2	R-152	I-101	11.5	29.1	40.1	17	15.4	39.5	52.3	9
	ToCo ($\lambda = 0$)	✓	1-2	R-152	I-101	11.4	29.9	41.0	18	14.9	41.4	57.2	7.5
	ToCo ($\lambda = 0.5$)	✓	1-2	R-152	I-101	**12.6**	**30.9**	**42.0**	**16**	**17.6**	**45.5**	**59.0**	**7**
I3D-Resnet152	MIL-NCE-t (Miech et al., 2020)	✗	n/a	n/a	I-152	**11.4**	**30.6**	**42.0**	**16**	**9.4**	**22.2**	30.0	**35**
	ActBERT (Zhu & Yang, 2020)	✗	0-12	O-101	R(2+1)D	9.6	26.7	**38.0**	**19**	**8.6**	**23.4**	**33.1**	**36**
	ToCo ($\lambda = 0.0$)	✗	1-2	R-152	I-152	9.6	25.0	34.3	27	4.3	12.8	21.4	72
	ToCo ($\lambda = 0.5$)	✗	1-2	R-152	I-152	**10.9**	**27.7**	37.2	23	4.9	14.7	23.4	60
	ToCo ($\lambda = 0$)	✓	1-2	R-152	I-152	11.7	28.3	37.6	21	19.3	44.7	59.3	7
	ToCo ($\lambda = 0.5$)	✓	1-2	R-152	I-152	**16.0**	**36.7**	**48.5**	**11**	**19.5**	**46.6**	**60.6**	**6**
S3D-HowTo100M	MIL-NCE-t (Miech et al., 2020)	✗	n/a	n/a	S-100	15.1	**38.0**	**51.2**	**10**	**9.9**	**24.0**	**32.4**	**29.5**
	MIL-NCE-g (Miech et al., 2020)	✗	n/a	n/a	S-100	8.8	24.3	34.6	23	**8.2**	**21.5**	**29.5**	**40**
	ToCo ($\lambda = 0$)	✗	2-2	n/a	S-100	13.7	32.0	43.4	15	6.1	16.5	23.1	62.5
	ToCo ($\lambda = 0.5$)	✗	2-2	n/a	S-100	**15.4**	**33.8**	**44.9**	**14**	6.5	18.5	25.0	56.5
	UniVL-v3 (Luo et al., 2020)	✓	6-2	n/a	S-100	**28.9**	**57.6**	**70.0**	**4**	21.2	49.6	63.1	6
	ToCo ($\lambda = 0$)	✓	2-2	n/a	S-100	24.3	51.2	64.4	**5**	**21.5**	48.4	61.6	6
	ToCo ($\lambda = 0.5$)	✓	2-2	n/a	S-100	**26.1**	**53.4**	**66.3**	**5**	20.8	**50.3**	**64.3**	**5**

Table 2: A complete comparison of **ToCo** under zero-shot(✗) and fine-tuned(✓) evaluation paradigms, with ($\lambda = 0.5$) and without ($\lambda = 0$) token-NCE, using three video representations, on both the YouCook2 and MSR-VTT retrieval benchmarks. For completeness, we also include comparisons to work with larger models and compute in gray.

5.2 Pre-training Results

Next we explore the role of **ToCo** when pretraining is included. This is made possible by the HowTo100M dataset of Miech et al. (2019). Pretraining introduces two evaluation settings: (1) Zero-shot, and (2) Fine-tuned. Both settings are presented in Table 2, again broken up by feature representation and comparing both our baseline model ($\lambda = 0$) and **ToCo**-balanced ($\lambda = 0.5$).

Zero-shot In the zero-shot setting, no in-domain training data is used from the task. Instead, the representations and scores built during pretraining are directly applied to the evaluation to test to their generalization. This leads to a number of domain shift issues. First, while all the videos originated on YouTube, the downstream task is curated and annotated. This means the quality of the videos may be different, the specific topics narrow, and the language includes full punctuated sentences. Second, the clips differ in length (both visual frames and linguistic tokens). Despite this, it is important to test the generality of the approach so all rows marked with ✗ compare our performance to other zero-shot results when available. In addition to the primary retrieval benchmarks we also

Method	Recall
Alayrac et al. (2016)	13.3
Zhukov et al. (2019)	22.4
Supervised (Zhukov et al., 2019)	31.6
TVJE (Miech et al., 2019)	33.6
MIL-NCE (Miech et al., 2020)	40.5
ActBert Zhu & Yang (2020)	41.4
UniVL-v3 (Luo et al., 2020)	42.0
Ours	**42.3**

Table 3: Average recall for action localization on CrossTask.

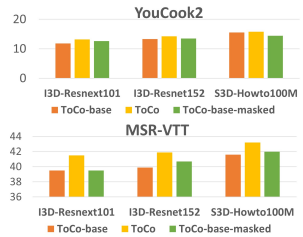


Figure 4: Ave. Recall@{1,5,10} for different features. ToCo-base-masked uses MLM.

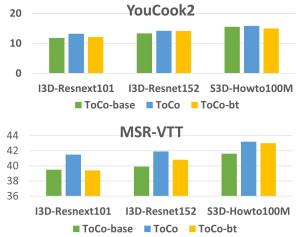


Figure 5: Ave. recall@{1,5,10} with λ=0 vs two variants of weighting: noun/verb/adj vs det/adp/aux.

present results on the task of Action Localization on the CrossTask dataset in Table 3. As we can see, our approach achieved the best recall though it merely uses two layer of video encoders.

Fine-tuning. In contrast, fine-tuning, allows us to continue training on in-domain data. For both pretraining and fine-tuning, it is unclear in general when to stop training, which is likely a function of the amount of data available (Gururangan et al., 2020). In our setup, we consistently fine-tune for 100k iterations with initial learning rate $1e^{-5}$ before evaluation. Rows with a ✓ include fine-tuning.

Table 2 presents a comprehensive set of results across three feature representations, zero-shot and fine-tuning, for both the YouCook2 and MSR-VTT datasets. We see that in the first setting (I3D-101) **ToCo** outperforms all models on both datasets with and without fine-tuning. In the richer feature domains **ToCo** performs competitively or outperforms comparable models. performs competitively or outperforms comparable models.

Model Complexity and Computational Resources As noted previously, results in this domain conflate feature representations, pretraining, and loss functions. We have therefore tried to provide coverage of these comparisons, but compute and model size are additional, important factors, we leave to future work. For completeness, we include two such innovations in our full table of results. Specifically, the concurrent and larger UniVL-v3 model of (Luo et al., 2020), and the TPU based end-to-end training of MIL-NCE-t (Miech et al., 2020). Specifically, UniVL-v3 uses three times as many video encoding layers which proved prohibitive on our hardware. Relatedly, MIL-NCE-t uses TPUs to perform full end-to-end feature fine-tuning with a large batch size of 8192. In contrast, **ToCo** was trained with batches of 64 and precomputed features.

5.3 Inspecting the Loss

We further probe the proposed **ToCo** from two angles: (1) against MLM and, (2) syntactic class.

Masked Language Prediction. In this experiment, we replace **ToCo** loss with the masked language prediction loss (**ToCo**-base-masked), and then train on YouCook2 and MSR-VTT. As shown in Fig. 4, masked language prediction under-performs **ToCo** under different settings. Along with the comparisons above, we believe **ToCo** has learned a better grounding between text to video.

Token type matters. As we mentioned above, we extract the *noun*, *verb* and *adjective* to impose our **ToCo** loss. Fro comparison, we instead add our loss on top of the complementary tokens in a sentence, such as *adposition* and *determiner*, whcih is denoted by **ToCo**-bt In Fig. 5, we can see that adding our loss on the tokens which have meaningful grounding on the video contents is better.

6 Conclusion

In this work we introduce **ToCo** as an alternative training paradigm for cross-modal learning, and a nearly exhaustive comparison of its use on a simple, vanilla transformer to achieve competitive or state-of-the-art results on standard benchmarks. There are several natural open questions. We expect that the complementary nature of our contribution to advances in features, end-to-end training, or large scale modeling will allow it to function as a drop-in replacement to improve existing techniques. We anticipate similar trends to hold within images as well.

REFERENCES

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4575–4583, 2016.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. 05 2015. URL http://arxiv.org/abs/1505.00468.

T. Baltrušaitis, C. Ahuja, and L. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2019.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience grounds language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 05 2017. URL https://arxiv.org/abs/1705.07750.

Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A Short Note on the Kinetics-700 Human Action Dataset. In *arXiv admin note: substantial text overlap with arXiv:1808.01340*, 07 2019. URL https://arxiv.org/abs/1907.06987.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 10 2019. URL https://arxiv.org/abs/1810.04805.

Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9346–9355, 2019.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL https://www.aclweb.org/anthology/2020.acl-main.740.

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6546–6555, 2018.

Stevan Harnad. The symbol grounding problem. pp. 1–22, 03 1990.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 12 2015. URL https://arxiv.org/abs/1512.03385.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2758–2766, 2017.

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset. *arXiv:1705.06950*, 05 2017. URL https://arxiv.org/abs/1705.06950.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. URL https://arxiv.org/abs/1602.07332.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 201–216, 2018.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020a.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. *arXiv preprint arXiv:2004.06165*, 2020b.

Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2657–2664, 2014a.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014b.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances in neural information processing systems*, pp. 289–297, 2016.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *11 pages, 5 figures*, 08 2019. URL https://arxiv.org/abs/1908.02265.

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

A. Miech, J. B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9876–9886, June 2020. URL https://arxiv.org/abs/1912.06430.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 06 2019. URL https://arxiv.org/abs/1906.03327.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013. URL http://arxiv.org/abs/1301.3781.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. Videobert: A joint model for video and language representation learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7463–7472, 2019.

Hao Tan and Mohit Bansal. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *EMNLP*, 08 2019. URL https://arxiv.org/abs/1908.07490.

Jesse Thomason, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Raymond Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 1218–1227, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C14-1115.

Atousa Torabi, Niket Tandon, and Leonid Sigal. Learning language-visual embedding for movie understanding with natural-language. *arXiv preprint arXiv:1609.08124*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 06 2017. URL https://arxiv.org/abs/1706.03762.

Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 (2):394–407, 2018.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5288–5296. IEEE, 2016. ISBN 978-1-4673-8851-1. URL http://ieeexplore.ieee.org/document/7780940/.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 21–29, 2016.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4651–4659, 2016.

Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. Video captioning and retrieval models with semantic attention. *arXiv preprint arXiv:1610.02947*, 6(7), 2016.

Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3165–3173, 2017.

Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 471–487, 2018.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6720–6731, 2019.

Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI 2018*, 2017.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8739–8748, 2018.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2019. URL `https://arxiv.org/abs/1909.11059`.

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

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3537–3545, 2019.

A Appendix

A.1 Contrastive Loss

The joint probability $p(\boldsymbol{x}_i, \boldsymbol{y}_j)$ for a video-text pair $(\boldsymbol{x}_i, \boldsymbol{y}_j)$ is approximated by:

$$p(\boldsymbol{x}_i, \boldsymbol{y}_j) \sim \frac{\exp^{f(\boldsymbol{x}_i, \boldsymbol{y}_j)}}{\sum_{k=1}^{N} \sum_{l=1}^{N} \exp^{f(\boldsymbol{x}_k, \boldsymbol{y}_l)}} \tag{4}$$

Based on equation 4, estimating a good joint probability is equivalent to the learning of a good joint representation f, such that $p(\boldsymbol{x}_i, \boldsymbol{y}_i) > p(\boldsymbol{x}_j, \boldsymbol{y}_i), \forall j \neq i$ and $p(\boldsymbol{x}_i, \boldsymbol{y}_i) > p(\boldsymbol{x}_i, \boldsymbol{y}_j), \forall j \neq i$. In the following, we describe how a contrastive loss can be derived to learn the desired joint representations, and how our proposed token-wise weighting is applied.

The denominator of equation 4 requires an intractable sum over all possible video-text pairs. Note that $p(\boldsymbol{x}_i, \boldsymbol{y}_i) > p(\boldsymbol{x}_j, \boldsymbol{y}_i), \forall j \neq i$ is identical to $p(\boldsymbol{x}_i|\boldsymbol{y}_i) > p(\boldsymbol{x}_j|\boldsymbol{y}_i), \forall j \neq i$ according to Bayes' theorem. Hence, we instead estimate the conditional probabilities $p(\boldsymbol{x}_j|\boldsymbol{y}_i)$ and $p(\boldsymbol{x}_i|\boldsymbol{y}_j)$. Where:

$$p(\boldsymbol{x}_j|\boldsymbol{y}_i) = \frac{p(\boldsymbol{x}_j, \boldsymbol{y}_i)}{p(\boldsymbol{y}_i)} \sim \frac{p(\boldsymbol{x}_j, \boldsymbol{y}_i)}{\sum_{k=1}^{N} p(\boldsymbol{x}_k, \boldsymbol{y}_i)} \tag{5}$$

where $\sum_{k=1}^{N} p(\boldsymbol{x}_k, \boldsymbol{y}_i)$ is an approximation of the marginal distribution $p(\boldsymbol{y}_i)$. Substituting equation 4 into equation 5, the denominator in equation 4 cancels, leaving:

$$p(\boldsymbol{x}_j|\boldsymbol{y}_i) \sim \frac{\exp^{f(\boldsymbol{x}_j, \boldsymbol{y}_i)}}{\exp^{f(\boldsymbol{x}_j, \boldsymbol{y}_i)} + \sum_{k \neq j} \exp^{f(\boldsymbol{x}_k, \boldsymbol{y}_i)}} \tag{6}$$