Cali-NCE: Boosting Cross-modal Video Representation Learning with Calibrated Alignment

Nanxuan Zhao  
Department of Computer Science  
University of Bath  
nanxuanzhao@gmail.com

Jianbo Jiao  
School of Computer Science  
University of Birmingham  
j.jiao@bham.ac.uk

Weidi Xie  
Cooperative Medianet Innovation Center  
Shanghai Jiaotong University  
weidi@sjtu.edu.cn

Dahua Lin  
Department of Information Engineering  
Chinese University of Hong Kong  
dhlin@ie.cuhk.edu.hk

Abstract

With the large-scale video-text datasets being collected, learning general visual-textual representation has gained increasing attention. While recent methods are designed with the assumption that the alt-text description naturally conveys the meaning and context of the video in semantics (i.e. well aligned with each other), it is unlikely to be satisfied for the Internet data, which potentially harms the quality of the learned visual-textual representation. To address this challenge, we first revisit three mainstream approaches: correspondence modeling, contrastive learning and predictive coding, demonstrating that a simple co-training strategy with these methods leads to a clear improvement in performance. To further explore the complementary nature of different training strategies, we propose a simple yet effective joint training framework that factorizes the total objective into conditional ones, termed as Cali-NCE\(^1\). Our method first estimates confidence scores for measuring the correspondence between video and text descriptions, and the scores are later used to calibrate the sample weightings during contrastive training. Through extensive experiments, we show that the proposed approach achieves state-of-the-art performance on multiple downstream tasks: text-to-video retrieval, video action recognition, and video retrieval.

1. Introduction

The highly developed intelligence of humans cannot be simply separated from vision and language [14, 15, 29]. They play key roles in our daily communication and how people understand the dynamic visual world. Modeling these means of communication becomes an important way also in the development of machine intelligence. A series of works have explored the modeling on such cross-modal tasks including text-to-video retrieval [41, 67], video captioning [31, 63, 74], and video question answering [3, 39]. In order to learn a good representation, previous works mostly rely on a set of well-annotated pairs of video clips and text fragments for individual tasks. The expensive and tedious annotation process limits the progress of general visual-textual representations.

Taking advantage of the rich video resources from online repositories (e.g. YouTube and Shutterstock), recent works [9, 45] contribute datasets on large scales, showing the potential of general visual-textual representation power on several downstream tasks. For instance, the HowTo100M dataset [45] consists of over 100 million video clips and associated narration pairs; the WebVid-2M [9] dataset contains more than two million video alt-text pairs. Driven by the great success of instance discrimination on general visual representation learning [13, 28, 70], the leading methods follow a common approach to align the video and text into a shared embedding space by pulling positive visual-textual feature pairs close to each other [44, 45], while the negatives apart.

However, these methods rely on the assumption that all video and alt-text pairs are well aligned in semantics, which cannot always be true for web data. Many text descriptions are vague and inaccurate because of the uncurated collection process and underlying ambiguities. For example, as shown in Fig. 1, the scene in (a) can also appear in other cities than London, such as Hong Kong; the scene in (b) is hard to tell as a viaduct. Instead, the samples in (c) and (d)
are more specific and semantically aligned, which should be
taken as more reliable supervision signals. This motivates
us to think about whether there exists a better way to model
the correlation between noisy web visual-textual pairs for
representation learning.

To address this problem, we first revisit three mainstream
methods for cross-modal representation learning, including
correspondence modeling [4, 5, 7], contrastive learning [1, 44],
and predictive coding [25, 56]. We find that these
methods serve for learning different aspects and are com-
plemented with each other, as a simple co-training can in-
crease the performance clearly. Based on the observations,
we propose a new method named Cali-NCE by introducing
 calibrated alignments in the co-training framework of cor-
respondence modeling and contrastive learning. Cali-NCE
uses the predicted confidences from correspondence mod-
eling for calibrating the supervision signal of each visual-
textual pair during training.

To validate the effectiveness of the proposed method,
we conduct extensive experiments on multiple challeng-
ing downstream tasks: zero-shot text-to-video retrieval,
video action recognition, and video retrieval. The proposed
method achieves state-of-the-art performance on all these
tasks. In summary, we make the following contributions:

• We take a step forward to investigate the problem
of noisy semantic alignments for cross-modal visual-
textual representation learning. We probe different
approaches including correspondence modeling, con-
trastive learning and predictive coding, showing their
potential for the target problem.

• We propose a new joint training framework with both
correspondence prediction and contrastive learning. In
particular, we introduce a new Calibrated Noise Con-
trastive Estimation (Cali-NCE) loss based on the pre-
picted correspondence confidence score.

• We evaluate the quality of the learned representa-
tions and design choices through extensive experi-
ments across different downstream tasks over several
datasets with noticeable improvement, and achieve
state-of-the-art performance.

2. Related Work

Learning from visual and textual cues has gained a sig-
nificant amount of attention and attracted many works on
related applications [3, 31, 41, 67, 74]. In this section, we
focus on the major related works and the most relevant pa-
pers.

Video self-supervised learning. Self-supervised learn-
ing aims to learn semantically meaningful representations
by generating supervision signals from the videos them-
selves. It has gained more and more attention as a good
representation can benefit many downstream tasks. At
the earlier stage, a set of pretext tasks are designed ar-
tificially based on temporal or spatial information, such
as identifying the odd video sequence [18], sorting se-
quences [37], predicting the arrow of time [68], clip order
prediction [72], and rotation modeling [30]. With the im-
pressive results generated by contrastive loss on ImageNet
pretrained dataset [8, 28, 70], the contrastive learning also
adapts to video unsupervised learning [54]. This kind of
method pulls two views of a sampled instance together, and
pushes away from the other instances.

Multimodal representation learning. Video data ac-
company with multiple modalities, such as visual images,
audio, text, and motion. Leveraging this multi-modality
nature can enhance the video representation and has been
explored by many recent works. [27] learn to associate
images with spoken words, and [7] directly model video,
audio, and text at the same, relying on a curated anno-
tation dataset. Instead, [7] train a multimodal versatile
model without any human label. Among different multi-
modal modelings, video and audio is one of the biggest
branches. A set of works [4, 5, 35, 50] exploit the video-
audio co-occurrence for learning a good video representa-
tion. [4] design an audio-visual correspondence (AVL) task
by classifying whether a pair of video and audio clips is
 corresponded with each other. And they further extend this
idea and design a model [5] that can work for both cross-
modal retrieval and sound source localization. [2] introduce
a cross-modal deep clustering method by supervising one
modality by the cluster of the other modality.

Vision and language. Researches in this area adopt a
common approach by embedding visual and textual repre-
sentation into the same space [19, 69], and use the dis-
tance (e.g. a dot product) to measure the semantic sim-
ilarity across modalities. Learning such a joint visual-
textual space has shown its effectiveness among many
works [34, 51, 66]. But most of these works rely on the
medium-scale well-annotated datasets, which limits the
scalability of the learned representation. With the recent
release of large-scale datasets collected from the Internet
without much human intervention [9, 45, 57], many recent
works have appeared to study visual-textural representa-
tion learning [16, 21, 38, 53, 60, 78]. Different from previ-
ous works [26, 44] studying the temporal misalignment, our
work tackles the misalignment in semantics.

3. Revisiting Cross-model Representation
Learning Methods

While contrastive learning is the main method used for
visual-textual representation learning with large datasets,
how to model cross-modal representations in other modal-
ities has been studied for a long run [4, 5, 56]. Inspired by
this, rather than directly working on contrastive learning, in
this section, we first study three general cross-modal representation learning methods, including correspondence modeling, contrastive learning, and predictive coding.

3.1. Visual-textual Correspondence Modeling

While watching movies, or instructional videos, the texts/subtitles often come with the material. To leverage this information, the visual-textual correspondence modeling is designed as a binary classification task, that is, to predict whether a video clip and the description text are corresponded or not (Fig. 2 (a)). During training, we treat the video clip and the text taken at the same time as the corresponding pair, while the video clip with texts coming from other randomly sampled videos as non-corresponding pairs. More specifically, given a video clip $v_i$ with its corresponding text $t_i$, we first extract the feature embedding $f_{v_i} \in \mathbb{R}^{d_v}, f_{t_i} \in \mathbb{R}^{d_t}$ using a visual backbone and textual backbone, respectively. After concatenating these two features together, a correspondence modeling sub-network $C$ is derived to obtain the correspondence confidence value $\hat{c}_{ij}$ within a range of $(0, 1)$:

$$\hat{c}_{ij} = C([f_{v_i}, f_{t_i}]),$$

where $[*,*]$ is a concatenation operation, and the ground-truth value $c_{ij}$ is set to 1. Similarly, we obtain the correspondence confidence for an unmatched pair $v_i$ and $t_j$ as:

$$\hat{c}_{ij} = C([f_{v_i}, f_{t_j}]),$$

where the ground-truth label $c_{ij}$ is set to 0. Then the correspondence loss is defined as a binary cross entropy loss:

$$L_{corr} = -c_{ij} \log (\hat{c}_{ij}) - (1 - c_{ij}) \log (1 - \hat{c}_{ij}). \quad (1)$$

Although the formulation is simple, learning to predict correspondence is not a trivial task. It requires a thorough understanding of the visual concepts within the videos to match with the semantic meaning conveyed by the textual descriptions. Note that there are no explicit cross-modal distance constraints imposed on the formulation, more freedom is provided to the learning process and the learned representation has the potential to be useful for more downstream tasks other than retrieval.

3.2. Visual-textual Contrastive Learning

Contrastive learning [12, 49] has prevailed in representation learning in many domains, such as vision-only, visual-audio, and also visual-textual, because of its flexibility in loss design and superiority in performance. There are several variants [1, 44], but here we mainly describe the commonly used version. The goal of contrastive learning is to learn a joint embedding space $E$ where the corresponded video and text features $f_{v_i}, f_{t_i}$ are close to each other, and far away from each other if not corresponding. Besides, the semantic comparisons between the two modalities can be made by simple dot products in the embedding space [1, 44]. By assuming that the joint probability can be estimated up to a constant factor over the exponentiation of the dot product of two feature embeddings, we obtain:

$$E(v_i, t_i) \propto \exp(f_{v_i}^\top f_{t_i}).$$

Accordingly, we adopt the NCE loss [1, 44] for optimization by differentiating between data obtained from the true joint distribution $E$ and some artificially generated negative data. For a sample $v_i$, the positive training pair is the same as those generated in Section 3.1 that are taken at the same position. The negative pairs are obtained from different videos. For example, if a mini-batch contains $N$ samples, then there will be $N$ positive pairs and $N^2 - N$ negative pairs. Similarly, for a sample $t_i$, the training pairs are generated in the same manner. We thus define the NCE loss $NCE(v_i, t_i)$ and the loss for contrastive learning $L_{cont}$ as:

$$NCE(v_i, t_i) = -\log \frac{\exp(f_{v_i}^\top f_{t_i}/\tau)}{\exp(f_{v_i}^\top f_{t_i}/\tau) + \sum_{j \in \mathcal{N}_i} \exp(f_{v_i}^\top f_{t_j}/\tau)},$$

$$L_{cont} = NCE(v_i, t_i) + NCE(t_i, v_i), \quad (2)$$

where $\mathcal{N}_i$ denotes the negative index sets for the sample $*$ and $\tau$ is a temperature parameter.

3.3. Visual-textual Predictive Coding

Compared with correspondence modeling, contrastive learning can be regarded as a stricter method, with explicit
(a) Correspondence Modeling

(b) Contrastive Learning

(c) Predictive Coding

Figure 2. Cross-modal representation learning methods examined for our problem. (a) Correspondence modeling aims to classify whether a pair of video and text clip is corresponded or not, i.e., binary classification. (b) Contrastive learning aims to pull positive pairs together while negative pairs apart in the embedding space. (c) Predictive coding aims to predict the embeddings of the corresponding modality \( A \) based on the embeddings of modality \( B \) \( (A, B \in \{ \text{video}, \text{text} \}) \) in this work.

Table 1. Analysis of different methods on MSR-VTT zero-shot text-to-video retrieval task. \( \text{R}@K \): Recall@K. \( \text{MedR} \): Median Rank. \text{Corr.:} Correspondence modeling. \text{Contra.:} Contrastive learning. \text{Coding.:} Predictive Coding.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \text{R}@1 \uparrow )</th>
<th>( \text{R}@5 \uparrow )</th>
<th>( \text{R}@10 \uparrow )</th>
<th>\text{MedR} \downarrow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.01</td>
<td>0.05</td>
<td>0.1</td>
<td>500</td>
</tr>
<tr>
<td>Coding.</td>
<td>0.4</td>
<td>1.9</td>
<td>4.8</td>
<td>180.0</td>
</tr>
<tr>
<td>Corr.</td>
<td>2.4</td>
<td>7.3</td>
<td>11.0</td>
<td>141.0</td>
</tr>
<tr>
<td>Contra.</td>
<td>10.1</td>
<td>22.9</td>
<td>31.5</td>
<td>36.0</td>
</tr>
<tr>
<td>Contra. + Corr.</td>
<td>11.1</td>
<td>26.0</td>
<td>35.9</td>
<td>28.0</td>
</tr>
<tr>
<td>Contra. + Coding.</td>
<td>11.6</td>
<td>26.0</td>
<td>34.8</td>
<td>28.0</td>
</tr>
<tr>
<td>Contra. + Corr. + Coding.</td>
<td>11.5</td>
<td>26.2</td>
<td>34.8</td>
<td>27.0</td>
</tr>
</tbody>
</table>

constraints imposed on the dot product of the cross-model features. To step further, in this subsection, we introduce the visual-textual predictive coding, by directly regressing the predicted coding of the other modality. This regression formulation is simpler than the above two methods, as it does not require large batches for generating negative sample pairs, and has been validated in prior works [25, 56] on representation learning of other domains. The assumption is that a good representation should maintain useful semantic features to accomplish this task.

Given a video clip \( v_i \) and its in-sync text \( t_i \) as the inputs, we treat the features \( f_{vi}, f_{ti} \), extracted from backbones as the coding for prediction. A small multi-layer perceptron (MLP) header is added to each backbone for the predictive coding generation, denoted as \( G_v, G_t \) for video and text respectively. The training loss then contains two components:

\[
L_{pred} = L_{v\rightarrow t}(v_i, t_i) + L_{t\rightarrow v}(v_i, t_i). \tag{3}
\]

We use the mean square error (MSE) loss for predictive coding, with the detailed loss defined as:

\[
L_{v\rightarrow t}(v_i, t_i) = \frac{1}{2} \left( \frac{G_v(f_{vi})}{\| \frac{f_{vi}}{\| f_{vi} \|} \|} - sg \left( \frac{f_{ti}}{\| f_{ti} \|} \right) \right)^2, \tag{4}
\]

\[
L_{t\rightarrow v}(v_i, t_i) = \frac{1}{2} \left( \frac{G_t(f_{ti})}{\| \frac{f_{ti}}{\| f_{ti} \|} \|} - sg \left( \frac{f_{vi}}{\| f_{vi} \|} \right) \right)^2, \tag{5}
\]

where \( sg[x] \) denotes the “stop gradient” operation to avoid the model collapse [23, 56].

3.4. Analyses

Implementation details. For the video branch, we use the standard S3D implementation following previous works [44, 71] throughout all the experiments. The video clip is sampled at 5 fps with 16 frames \( (i.e. 3.2 \text{ seconds}) \). Each frame is resized to a resolution of 224 \( \times \) 224. For the text branch, we use the word2vec embedding \( (d = 300) \) pretrained on Google News in a self-supervised manner [46] followed by two linear layers with a max-pooling layer in between [44]. For each text input, the maximum number of words is set to 30. The dimensions of both video and text features in the embedding space are 512. For the correspondence subnet \( C \), we follow the work [4] using a two-layer MLP and output a 2-dimensional softmax vector. For the coding subnet \( G \), we use a linear layer with the same dimension on input and output for prediction. We train our model on WebVid-2M [9] which contains 2.5M video-text pairs scraped from the web. The dataset includes a variety of styles for description text and is one of the most recently released web datasets. We compare the performance on MSR-VTT zero-shot text-to-video retrieval task to evaluate the learned cross-modal representation.

The result is shown in Table 1. Though correspondence modeling has demonstrated its efficiency in visual-audio representation learning, training only with correspondence loss degrades the performance a lot, especially com-
pared with the model trained with contrastive loss. We find that the training loss of correspondence classification converges very quickly. This may be due to the task of judging whether a video-text pair is corresponding is relatively easier. In this way, the model may overfit to the pretrained task without learning a more useful representation for the downstream task. Instead, by contrasting with a set of negative pairs and adding constraints on the dot production across modalities, contrastive learning has shown its superiority. This also aligns with the conclusion drawn from previous works [1, 9, 45]. For predictive coding, the performance is even worse than correspondence modeling. One possible reason is that there still has a certain level of collapse although we adopt the stop gradient during training. We encourage future research to further explore this direction.

We then try on different combinations of methods. The loss function is defined as $L = \theta_1 L_{corr} + \theta_2 L_{cont} + \theta_3 L_{pred}$. We set $\theta_1 = \theta_2 = 1$ and $\theta_3 = 200$ to balance the losses if it is used in the training. For example, Contra. + Corr. means we jointly train correspondence and contrastive together with $L_{corr} + L_{cont}$. We treat contrastive learning as a basic model because of its superiority in performance. It works surprisingly well when combining either correspondence classification or coding prediction with contrastive learning. This indicates that other methods could complement with contrastive learning, and further enhance the representation quality. Whereas adding all of these methods together does not explicitly boost the performance. We thus choose correspondence modeling to complement with contrastive learning and propose our Cali-NCE next.

### 4. Cali-NCE with Correspondence Modeling

In this section, we further explore the complementary information between correspondence modeling and contrastive learning. According to the instance discrimination assumption, samples extracted from the other videos are all treated as negative ones. However, among these negatives, some of them can be false negatives. For example, there may exist two different videos (i.e. $v_a, v_b$) both describing “slip the stitch to the right needle” (i.e. for texts $t_a, t_b$, and $t_a = t_b$ in this case), and contrastive learning will treat $v_a, t_b$ as negative pairs, which is apparently not the reliable supervision signal (Fig. 3 (a)). Without posing any constraints on negative samples, joint training with correspondence tasks can alleviate this false negative issue. The reason behind this is that it is much easier for the model to pull such two videos together in the embedding space to fit the correspondence modeling goal (i.e. binary classification) than repelling them apart.

On the other hand, false positives also exist in the data samples, especially for the pretraining data that are not carefully curated/annotated. Except the examples shown in Fig. 1, the sample video clip and the text taken at the same time may not be aligned well [44]. For example, when watching an NBA game, the description text comes out only after the goal. In Fig. 3 (b), the misalignment comes out because of the shooting way, and the video producer may add some irrelevant clips for better storytelling and aesthetic pleasure. Directly increasing the dot product similarity among video and text pairs is inappropriate. Instead, the correspondence loss does not add on dot product similarity.
and provides model flexibility for training. Correspondence can be considered as a weak label, which is a more accurate supervision signal in this case.

Simply joint training with the correspondence modeling task is a very straightforward way to mitigate the above problems, and bring the performance gain (Section 3.4). But this is hard-coded and the affinity between modalities is uncontrollable. Here, based on the findings, in addition to joint training, we further propose a more adaptive solution Cali-NCE, by leveraging the correspondence prediction confidence. For each positive pair \((v_i, t_i)\), we first estimate its correspondence confidence value \(\hat{c}_{ii}\) via the correspondence modeling (Section 3.1). If the pair is more correlated, the confidence is larger (with an upper bound of 1). We then calibrate the contrastive loss by re-weighting with \(\hat{c}_{ii}\) as follows:

\[
L_{\text{CaliNCE}} = \lambda_1[\hat{c}_{ii} \ast (\text{NCE}(v_i, t_i) + \text{NCE}(t_i, v_i))] + \lambda_2 L_{\text{corr}},
\]

where \(\lambda_1\) is the weight used to balance the objective terms, and we set \(\lambda_1 = \lambda_2 = 1\) empirically throughout our experiments. Note that \(L_{\text{corr}}\) is necessary in Cali-NCE for estimating \(\hat{c}_{ii}\) in a data-driven and adaptive manner.

5. Experiments

Implementation details. We mainly follow the details mentioned in Section 3.4. Except for taking the convolutional neural network (CNN) as backbones, we also examine the Transformer-based backbone, which has shown great performance for representation learning tasks. We replace the standard S3D and MLP backbones with CLIP (ViT-B/32) [55] and extract features in 1 fps across 8 seconds. We freeze all the parameters from the CLIP model and use the features only. We build two Transformer layers with 512 hidden units and 8 heads on top of CLIP features for final evaluation on the downstream task.

5.1. Downstream Tasks

To show the effectiveness of our model and the generality of the learned representations, we conduct evaluations on three diverse downstream tasks: text-to-video retrieval, video action recognition, and video-to-video retrieval. We first introduce the downstream tasks with the corresponding datasets and required metrics below.

Text-to-video retrieval. This task aims to retrieve the best-matched video according to the input text. We focus on the zero-shot text-to-video retrieval as we want to directly evaluate the quality of the learned representation without further finetuning. And we use the dot product to measure the similarity across modalities. We test this task on two datasets: 1) MSR-VTT [73] is a dataset that contains 200K unique video clip-caption pairs covering 20 different categories. We use the same test split with 1K constructed by a previous work [75] that is commonly used in the literature [9, 10, 44]. 2) MSVD [11] contains 1,970 videos with 80K descriptions. We use the standard split with 670 videos in the test dataset [42]. We report the results on recall metrics (R@K, \(K = 1, 5, 10\)) which measure the percentage of correctly retrieved clips at the top \(K\). We also report the median rank (MedR) of videos to be retrieved.

Video action recognition. This task targets to classify the action class of each given video. We follow the released protocol [44] by training directly on pre-extracted features with a linear SVM. We test on both HMDB-51 [36] and UCF-101 [58] datasets and report the average accuracy over three splits.

Video-to-video retrieval. In this task, we evaluate the learned video representation in a zero-shot manner through video retrieval. Similarly, we examine on the most commonly used datasets: HMDB-51 [36] and UCF-101 [58]. Given a video, we uniformly sample 10 clips, each with 16 frames [72]. We obtain the final feature by averaging over these 10 clips. Then after extracting the feature from our model, we use cosine similarity to measure the distance between videos from the test set and those from the training split. The video retrieval performance is measured by querying the top \(K\)-nearest neighbors (NN) on the test split, where \(K\) is set to 1, 5, 10, 20, and 50. The retrieval result is considered as success if the class label of the test clip is within the top \(K\)-NN.

5.2. Comparison with State-of-the-art Methods

To better validate the effectiveness of the proposed Cali-NCE in a more controlled manner, here we design a baseline by training with vanilla NCE and keeping all the other settings the same as ours.

Zero-shot text-to-video retrieval. In Table 2, we evalu-
ate our learned representation on MSR-VTT. We achieve the state-of-the-art (SOTA) performance and outperforms contrastive learning-based models [44, 45] by a large margin. Our off-the-shelf model without finetuning models directly trained on MSR-VTT [33, 62, 75–77]. Additional results on MSVD can be found in Table 3. Though the performance gain is not as high as that on MSR-VTT, our model still achieves better performance than other methods. We attribute this to that MSVD is a rather simple task with little room for improvement. The above shown performance over different datasets validates the effectiveness of our learned visual-textual representations.

**Video retrieval.** To evaluate the learned representation quality from our video branch, we conduct a zero-shot video retrieval experiment as in Table 4, where we report the average recalls over three splits on both HMDB-51 and UCF-101. A baseline method as well as other SOTA video representation learning methods are included for the comparison (including both single-modal and multi-modal approaches). The proposed approach achieves comparable performance to SOTAs on HMDB-51 and outperforms them on UCF-101 by a large margin.

**Video action recognition.** We show the results of our model compared to baseline, and other alternative methods on the task of action recognition in Table 5. The accuracy is averaged over three splits on both HMDB-51 and UCF-101 datasets. We can see that our model surpasses baseline and other SOTAs on both datasets, putting more evidence on the effectiveness of our video-only representation learned from the cross-modal task. Though our model only achieves comparable results on video retrieval with HMDB-51, after a linear SVM, we surpass the SOTA methods with a notable gain, suggesting that our learned embedding is semantically-aware and separable.

### Table 3. Comparison to SOTA results on MSVD for text-to-video retrieval. Numbers of previous works for non zero-shot methods are copied from [9].

<table>
<thead>
<tr>
<th>Method</th>
<th>R@1↑</th>
<th>R@5↑</th>
<th>R@10↑</th>
<th>MedR↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSE-LSTM [33]</td>
<td>12.3</td>
<td>30.1</td>
<td>42.3</td>
<td>14.0</td>
</tr>
<tr>
<td>VSE++ [17]</td>
<td>15.4</td>
<td>39.6</td>
<td>53.0</td>
<td>9.0</td>
</tr>
<tr>
<td>Multi. Cues [48]</td>
<td>20.3</td>
<td>47.8</td>
<td>61.1</td>
<td>6.0</td>
</tr>
<tr>
<td>CE [42]</td>
<td>19.8</td>
<td>49.0</td>
<td>63.8</td>
<td>6.0</td>
</tr>
<tr>
<td>SupportSet [53]</td>
<td>23.0</td>
<td>52.8</td>
<td>65.8</td>
<td>5.0</td>
</tr>
<tr>
<td><strong>Zero-shot</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SupportSet [53]</td>
<td>21.4</td>
<td>46.2</td>
<td>57.7</td>
<td>6.0</td>
</tr>
<tr>
<td>MIL-NCE [44]</td>
<td>32.6</td>
<td>59.8</td>
<td>73.2</td>
<td>3.5</td>
</tr>
<tr>
<td>Baseline</td>
<td>26.6</td>
<td>54.6</td>
<td>66.9</td>
<td>5.0</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>32.8</td>
<td>63.0</td>
<td>76.4</td>
<td>3.0</td>
</tr>
</tbody>
</table>

### Video retrieval.

<table>
<thead>
<tr>
<th>Method</th>
<th>R@1</th>
<th>R@5</th>
<th>R@10</th>
<th>MedR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pace [65]</td>
<td>12.9</td>
<td>31.6</td>
<td>43.2</td>
<td>58.0</td>
</tr>
<tr>
<td>CoCLR [25]</td>
<td>26.1</td>
<td>45.8</td>
<td>-</td>
<td>69.7</td>
</tr>
<tr>
<td>SeLaVi [6]</td>
<td>24.8</td>
<td>47.6</td>
<td>-</td>
<td>75.5</td>
</tr>
<tr>
<td>GDT [52]</td>
<td>25.4</td>
<td>51.4</td>
<td>-</td>
<td>75.0</td>
</tr>
<tr>
<td>MIL-NCE [44]</td>
<td>32.9</td>
<td>52.3</td>
<td>61.1</td>
<td>70.5</td>
</tr>
<tr>
<td>Baseline</td>
<td>22.2</td>
<td>39.5</td>
<td>48.3</td>
<td>58.2</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>34.4</td>
<td>52.2</td>
<td>60.9</td>
<td>68.9</td>
</tr>
</tbody>
</table>

### Video action recognition.

<table>
<thead>
<tr>
<th>Method</th>
<th>HMDB51-Acc.</th>
<th>UCF101-Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPN [37]</td>
<td>23.8</td>
<td>59.6</td>
</tr>
<tr>
<td>ShuffleLearn [47]</td>
<td>35.8</td>
<td>68.7</td>
</tr>
<tr>
<td>MAS [64]</td>
<td>33.4</td>
<td>61.2</td>
</tr>
<tr>
<td>CMC [61]</td>
<td>26.7</td>
<td>59.1</td>
</tr>
<tr>
<td>Geometry [22]</td>
<td>23.3</td>
<td>55.1</td>
</tr>
<tr>
<td>Fernando et al. [18]</td>
<td>32.5</td>
<td>60.3</td>
</tr>
<tr>
<td>ClipOrder [72]</td>
<td>30.9</td>
<td>72.4</td>
</tr>
<tr>
<td>3DRotNet [30]</td>
<td>40.0</td>
<td>75.3</td>
</tr>
<tr>
<td>DPC [24]</td>
<td>35.7</td>
<td>75.7</td>
</tr>
<tr>
<td>Pace [65]</td>
<td>36.6</td>
<td>77.1</td>
</tr>
<tr>
<td>3D STPuzzle [32]</td>
<td>33.7</td>
<td>65.8</td>
</tr>
<tr>
<td>CBT [59]</td>
<td>44.6</td>
<td>79.5</td>
</tr>
<tr>
<td>MIL-NCE [44]</td>
<td>47.0</td>
<td>80.0</td>
</tr>
<tr>
<td>Baseline</td>
<td>46.1</td>
<td>76.3</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>49.6</td>
<td>81.4</td>
</tr>
</tbody>
</table>

### 5.3. Ablation Studies

**The effect of backbone.** Because of the great success of using Transformers [9, 38], we compare our method with SOTAs by using CLIP as the backbone. The results are shown in Table 6, and we copy the values for direct retrieval with CLIP features from the previous paper [43]. As can be seen, our model still works with a transformer-based backbone. As our method is agnostic to the underlying backbone, it has the potential to combine with more advanced
Figure 4. Example text-to-video retrieval results on MSR-VTT using our trained joint embedding. The input text is shown on the top of each group and the correctly retrieved video (GT) is in an orange box. The candidate retrieval set is the whole test set.

Table 6. Comparison to SOTA results using Transformer-based backbone on MSR-VTT zero-shot text-to-video retrieval task.

<table>
<thead>
<tr>
<th>Method</th>
<th>R@1↑</th>
<th>R@5↑</th>
<th>R@10↑</th>
<th>MedR ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frozen [9]</td>
<td>18.7</td>
<td>39.5</td>
<td>51.6</td>
<td>10.0</td>
</tr>
<tr>
<td>ALPRO [40]</td>
<td>24.1</td>
<td>44.7</td>
<td>55.4</td>
<td>8.0</td>
</tr>
<tr>
<td>VIOLET [20]</td>
<td>25.9</td>
<td>49.5</td>
<td>59.7</td>
<td>-</td>
</tr>
<tr>
<td>CLIP [55]</td>
<td>31.2</td>
<td>53.7</td>
<td>64.2</td>
<td>4.0</td>
</tr>
<tr>
<td>Baseline</td>
<td>32.2</td>
<td>55.9</td>
<td>64.1</td>
<td>4.0</td>
</tr>
<tr>
<td>Ours</td>
<td>32.9</td>
<td>56.5</td>
<td>66.8</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Table 7. The robustness of Cali-NCE with noisy training data. We show the random shuffle noisy rate on the left (e.g., 10% indicates that this amount of training pairs is unmatched).

<table>
<thead>
<tr>
<th>Method</th>
<th>R@1↑</th>
<th>R@5↑</th>
<th>R@10↑</th>
<th>MedR ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>31.0</td>
<td>55.2</td>
<td>65.0</td>
<td>4.0</td>
</tr>
<tr>
<td>Ours</td>
<td>31.0</td>
<td>55.8</td>
<td>65.4</td>
<td>4.0</td>
</tr>
<tr>
<td>Baseline</td>
<td>30.9</td>
<td>54.1</td>
<td>64.7</td>
<td>4.0</td>
</tr>
<tr>
<td>Ours</td>
<td>32.0</td>
<td>55.8</td>
<td>65.3</td>
<td>4.0</td>
</tr>
<tr>
<td>Baseline</td>
<td>31.4</td>
<td>53.2</td>
<td>63.0</td>
<td>5.0</td>
</tr>
<tr>
<td>Ours</td>
<td>31.6</td>
<td>54.3</td>
<td>64.2</td>
<td>4.0</td>
</tr>
</tbody>
</table>

network architectures to further enhance performance.

**The robustness of Cali-NCE.** To verify the robustness of our method in dealing with noisy training pairs, we conduct the experiment by deliberately injecting noises into the training data. We randomly shuffle a ratio of pairs in training data to make them unmatched. We run this experiment with the CLIP-based backbone same as the above on a randomly selected 1/6 WebVid-2M dataset. The results are shown in Table 7, testing on MSR-VTT zero-shot text-to-video retrieval task. With the help of Cali-NCE, the performance drop incurred by the noisy training data has been mitigated even for a random shuffle ratio of 50%, demonstrating the effectiveness of our Cali-NCE for dealing with noisy input data.

**Qualitative results.** Example results of text-to-video retrieval on MSR-VTT are shown in Fig. 4, and we regard vanilla NCE as the baseline. Note how our model can retrieve matched videos with the input text. Compared to vanilla NCE, rather than only focusing on a single concept in the input text (e.g., “car” in Fig. 4), our Cali-NCE pays attention to more fine-grained details (e.g., “car” and “cartoon”), supporting a deeper understanding of text semantics.

**6. Conclusion**

In this work, the main question we would like to address, is the noisy pairwise data used for learning the visual-textual representation. We have examined several methods, and found that the methods like correspondence modeling and predictive coding, can complement contrastive learning. Based on the findings, we proposed a new optimization solution called Cali-NCE, and showed its effectiveness over extensive quantitative and qualitative experiments. We hope that our work can inspire more following-up research on exploring semantic misalignment problems for cross-modal representation learning.

**References**


[51] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, pages 4594–4602, 2016. 2


[67] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, pages 5005–5013, 2016. 1, 2


