TEACHTEXT: CrossModal Generalized Distillation for Text-Video Retrieval Supplementary Material

Ioana Croitoru [1,2,*] Simion-Vlad Bogolin[1,2,*] Marius Leordeanu[2,3]
Hailin Jin[4] Andrew Zisserman[1] Samuel Albanie[1,5,†] Yang Liu[1,6,†]
[1]Visual Geometry Group, Univ. of Oxford [2]Inst. of Mathematics of the Romanian Academy [3]Univ. Politehnica of Bucharest
[4]Adobe Research [5]Dept. of Engineering, Univ. of Cambridge [6]Wangxuan Inst. of Computer Technology, Peking Univ.

In this supplementary material, we provide additional details on the video embeddings (Sec. 1) and text embeddings (Sec. 2) used in the main submission. We provide further details of Fig. 1 from the main paper (Sec. 3) as well as details on optimization (Sec. 4), modifications to the embedding pre-processing pipeline used in prior work (Sec. 5) and summaries of the datasets used (Sec. 6). Finally, we include additional ablations (Sec. 7) and a more comprehensive set of metrics for comparison with previous work, along with qualitative results (Sec. 8).

1. Video embeddings (experts) description

In this work, we used the set of pretrained experts considered by the authors of [21]. For completeness, we summarise here the manner in which these experts were extracted.

- Two form of action experts are used: *Action(KN)* and *Action(IG)*. The former is an I3D architecture trained on Kinetics [5], which produces 1024-dimensional embeddings from frame clips extracted at 25fps and center cropped to 224 pixels. The *Action(IG)* model is a 34-layer R(2+1)D model [38] that has ben trained on IG-65m [12]: it operates on frames extracted at 30 fps in clips of 8 at 112×112 pixel resolution.
- Two forms of object experts are used, named *Obj(IN)* and *Obj(IG)*. They are produced from frame-level embeddings extracted at 25fps. The *Obj(IN)* model consists of an SENet-154 backbone [15] which has been trained on ImageNet for image classification. *Obj(IG)* is formed from a ResNext-101 [43] extractor which was trained on Instagram data that was weakly labelled with hashtags [24]. For both models, frames are resized to 224×224 pixels.
- The face expert uses a ResNet50 [13] that has been trained for task of face classification on the VGGFace2 dataset [4], producing a 512-dimensional embedding for each detected face following detection.
- The audio expert is produced using the VGGish model, trained for audio classification on the YouTube-8m dataset and described by [14].
- The scene expert is a 2208-dim embedding that is extracted frames (at 25 fps) for a center crop of 224×224 pixels. The model, which is pretrained on Places365 [49], uses a DenseNet-161 [16] architecture.
- The speech expert is produced using the Google Cloud API (to transcribe the speech content).
- The OCR expert is a word2vec encoding [26] of text detected in frames using [23, 35].

1.1. Experts refinement – Modifying the Kinetics action recognition model

Apart from dropping the OCR and face experts as described in the main paper, one small modification we propose to the expert selection made by [21] is to replace the *Action(IG)* from an I3D model [5] to an R2P1D model [38] (matching the architecture *Action(IG)*) which has also been pretrained on IG-65m [12] and then finetuned on the Kinetics dataset [5].

2. Text embeddings description

We use several text embeddings. In addition to the Sec. 3 from the main paper, further technical details about each of them are given below:

- **mt_grovle** [2] is a "vision-sensitive" language embedding which is adapted from w2v using WordNet and an original vision-language graph built from Visual Genome [19]. The size of the final pre-trained embedding is 300.
- **OpenAI-GPT** [32] is a pre-trained text embedding which uses transformers [39] and language modeling on a large corpus (the Toronto Book Corpus) (the final model has 110M params). The size of the final pre-trained embedding is 768.

- **RoBERTa** [22] is a BERT-based embedding [8]. The model is trained longer with bigger batch size on more data, having 125M params. The size of the final pre-trained embedding is 768.

- **ALBERT** [20] is a lightweight modification to BERT [8] which overcomes some memory limitations, having 11M params. The size of the final pre-trained embedding is 768.

- **GPT2-large** [33] is a transformer-based [39] model trained on even more data (40 Gb of text) without any supervision, having 774M params. The size of the final pre-trained embedding is 1280.

- **GPT2-xl** [33] is similar to GPT2-large, but has more parameters (1558M params). The size of the final pre-trained embedding is 1600.

- **W2V** [26] is one of the most popular text embeddings used in vision tasks. It uses a neural network model to learn word representations. The size of the final pre-trained embedding is 300.

3. Further Details for Fig. 1

In Fig.1 from the main paper we highlight that the gain for a model that uses multiple text embeddings (last bar) is comparable with the gain of a model that uses multiple video modalities (middle bar), having as comparison a model that uses only one video modality (first bar). The first bar represents the CE [21] model trained with one video embedding, namely **Obj(IG)** (the performance of the model is 19.8±0.1 in geometric mean of R1-R5-R10). The second bar represents a CE model using 7 video modalities both for inference and training (the performance of the model is 24.4±0.1 in geometric mean of R1-R5-R10). In the third and final bar of the chart we present the performance of using three different text embeddings with TEACHTEXT at training, while using only one text embedding at inference time (the performance of the model is 30.4±0.0 in geometric mean of R1-R5-R10). All the numbers are presented after the modification of the pre-processing pipeline (please see Sec. 5 for further details). All the experts used by CE [21] are described in Sec.1.

4. Optimization setup

CE+ models are trained in Pytorch [30] using the Adam optimizer [17]. We use a learning rate of 0.001 and weight decay of 1E-5. When using a base architecture different to the proposed CE+, we use the same hyper-parameters as in the public codebase for the underlying method (CE[1] and MMT[2]). For MoEE, we use the re-implementation provided by the authors of the CE method [21].

5. Modification to pre-processing pipeline

During our preliminary analysis, we found out that some pretrained expert models produce embeddings that are fairly sensitive to jpeg compression artifacts. To address this, we re-extracted features from video frames densely extracted with minimal jpeg compression (corresponding to the use of ffmpeg [36] and the `-qscale:v 2` flag). In order to be fair in our comparisons, we apply this corrections everywhere. Due to this factor, we re-train MoEE [25] and CE [21] and report higher numbers.

6. Dataset details

To provide an extensive comparison we test our approach on seven video datasets that have been explored in recent works as benchmarks for the task of text-video retrieval. Next, we give details about all the datasets used.

MSRVTT [44] contains 10k videos, each having 20 captions. In order to test the retrieval performance, we report results on the official split which contains 2990 videos for the test split and 497 for validation, following the setup used in [21]. We perform most of our ablations on this split. To enable comparison with as many other methods as possible, we also report results on the 1k-A split as used in [11, 21, 31]. For this split, we report the performance after training 100 epochs. The split contains 1000 video candidates for testing and 9000 for training. We use the same candidates as defined in [21] which are used by all the other works [11, 31, 46], using each of the 20 captions associated to each video independently during evaluation and averaging performance across them.

MSVD [6] contains 80k English descriptions for a total of 1970 videos. We use the standard split of 1200 (training), 100 (validation) and 670 (testing) as used in other works [21, 40, 45]. The videos from MSVD do not have audio streams.

DiDeMo [1] contains 10464 videos sourced from a large-scale creative commons collection [37] and features moments of unedited, diverse content (concerts, sports, pets etc.). The dataset comprises 3-5 pairs of descriptions per video. We adopt the paragraph-video retrieval protocols used by [21, 47] and use splits corresponding to 8392 train, 1065 validation and 1004 test videos.

LSMDC [34] contains 118081 short video clips extracted from 202 movies. Each clip is described by a caption that is either extracted from the movie script or from transcribed DVS (descriptive video services) for the visually impaired. There are 7408 clips in the validation set and the testing is performed on 1000 videos from movies that

[1] https://shorturl.at/ksxIS

[2] https://github.com/gabeur/mmt

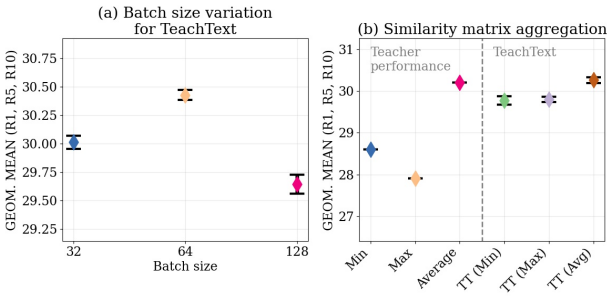

Figure 1. **(a) Batch size variation.** We vary the batch size for the MSR-VTT dataset to see how this affects the performance. We observe that batch size influences performance. The underlying architecture used for this experiment is CE+. **(b) Similarity matrix aggregation.** We present a comparison of different similarity matrix aggregation: *min*, *max* and *average*. As can be seen, the average aggregation has the best results (both when evaluating the teacher standalone or in conjunction with our TEACHTEXT algorithm).

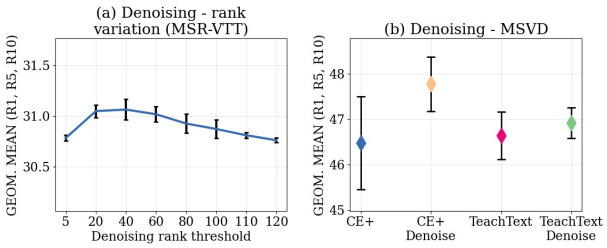

Figure 2. **(a) Rank variation for denoising.** The denoising involves dropping captions that are assigned a low ranking by the teacher for the training set. In this experiment, we vary the rank below which we drop sentences. Please note that for a rank of 5 (on the training set) the amount of dropped sentences is approximately 46%. Note that MSR-VTT has 20 captions per video, so after applying this filter we keep on average 10 captions per video. **(b) Denoising.** We present the effect of denoising on retrieval performance on MSVD. Some of the captions available in datasets with multiple captions per video may be noisy and actively harm the training process. We estimate the degree of noise present in a caption by looking at the teacher rank and drop the caption if necessary. We observe the effectiveness of denoising when applied in isolation (*CE+* vs *CE+ Denoise*) and in conjunction with the full TEACHTEXT method. The experiment is presented for dropping sentences that rank higher than rank 100.

are disjoint from the training and val sets as described in the Large Scale Movie Description Challenge (LSMDC)[3].

ActivityNet [3] contains 20k videos extracted from YouTube and has around 100K descriptive sentences. We follow the same paragraph-video retrieval setup as used in prior works [21, 47] and report results on the `val1` split. So, we use 10009 videos for training and 4917 videos for testing.

VaTeX [42] contains 34911 videos with multilingual captions (Chinese and English). There are 10 captions per video for each language. We follow the same protocol as in [7, 31] and split the validation set equally (1500 validation and 1500 testing videos). In this work, we only use the English annotations.

QuerYD [28] contains 1815 videos in the training split and 388 and 390 for validation and testing. The videos are sourced from YouTube and cover a diverse range of visual content. The dataset contains 31441 descriptions, from which 13019 are precisely localized in the video content (having start time and end time annotations) and the other 18422 are coarsely localized. For this work, we do not use the localization annotations and report results for the official splits.

7. Ablations

In this section, we present additional ablations.

7.1. Batch size variation

In Fig. 1a we vary the batch size for the MSR-VTT dataset in order to see how the performance is affected. As can be seen, we obtain the best value using the same batch size as for the method without applying TEACHTEXT algorithm (64 in this case).

7.2. Similarity matrix aggregation study

In Fig. 1b we present several similarity matrix aggregation possibilities: *min*, *max* and *average*. We observe that using the mean of the similarity matrices is more effective. Because of this, we use the mean as the final aggregation technique in our TEACHTEXT algorithm.

7.3. Denoising

In Fig. 2a we vary the threshold used to filter out sentences from the training set. As can be seen, this denoising method is effective and it can provide a significant gain in performance. In this experiment we have found out that the best threshold for MSRVTT is rank 40. Additionally, we present denoising results in Fig. 2b for the MSVD dataset using the 100 threshold. This method turns out to be effective in reducing noise for retrieval datasets. Denoising is not use in any other ablation studies. The final results when comparing with other state of the art methods are presented using denoising on MSRVTT and MSVD datasets.

7.4. Distillation setup

As stated in the main paper, the distillation setup admits a number of variants. In addition to the methods presented in Fig. 6b from the main paper, in Fig. 3a we present several additional comparisons. More exactly, we test our approach against a more classical distillation setup where we

[3] `https://shorturl.at/cdrI6`

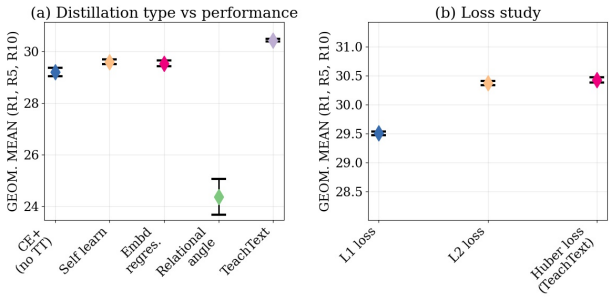

Figure 3. **(a) Distillation type.** The first bar represents the student performance without distillation (*CE+*). In addition to the methods presented in the main paper, here we test other distillation approaches: *Embd regress* which is a classical approach where the query and video joint embeddings are directly regressed based on the embeddings given by the teacher, *Relational angle* where we apply the angle relationships as introduced by [29]. In addition, we present results of our method in a self-learning setup where the teacher is the student from a previous run (*Self learn*). The last bar represent the performance of the TEACHTEXT approach. **(b) Loss study.** In this picture, we show how various distillation losses (L1, L2, Huber) affect the performance.

directly regress the embeddings given by the teacher (*Embd regress*). This setup does not follow the idea of relational distillation. Additionally, we also apply the angle distillation as introduced by [29] (*Relational angle*) where we use exactly the loss as in the public code[4]. Please note that the drop in performance as opposed to the student without distillation can be explained by some technical challenges that we encountered in order to make the angle loss compatible with the ranking loss used for this task. Last but not least, we also show that a small improvement can be obtained by using a self learning technique, where the teacher has the exact same architecture and inputs as the student.

7.5. Loss study

In the main paper, we follow recent literature [29] and use the Huber loss for distillation. However, we wanted to see how various losses affect the performance. We test with L1 and L2 losses. As can be seen in Fig. 3b, the Huber loss performs better than L1 loss and a bit better than L2.

7.6. Mixture of architectures

Our TEACHTEXT assumes that the only difference between the teacher and the student is the used pre-trained text embedding fed to the model. However, our method is not limited to this constraint. In this section, we show how having multiple teachers, that now have a different underlying architecture affect the performance of our method. Please note that in all other ablations, the architecture is shared between student and teacher. This is the only exception. Our preliminary results shown in Fig. 4 suggest

[4] https://github.com/lenscloth/RKD

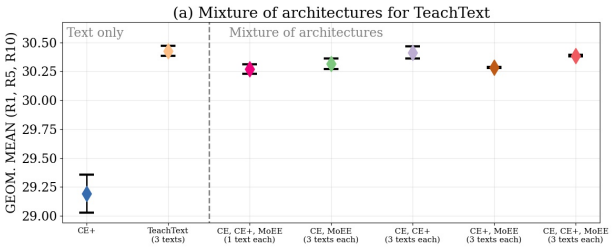

Figure 4. **Mixture of architectures.** We perform some preliminary experiments to see if the method may benefit from learning from teachers that do not share the same architecture. The x axis corresponds to the models which are used as teachers. In cases labeled with *3 text each*, we used three different variations of each architecture as teachers, accounting for a total number of *no. methods * 3* teachers. As can be seen, the results suggest that there is no clear benefit in using multiple architectures as teachers.

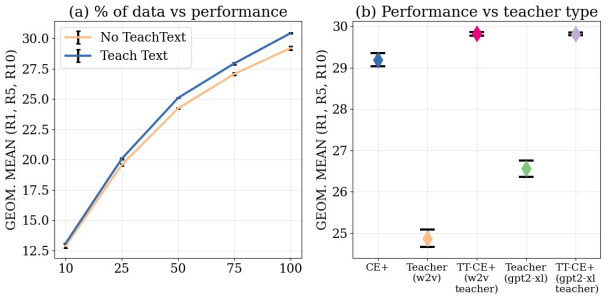

Figure 5. **(a) Amount of training data vs performance.** As it can be seen, with the increase of training data, the improvement brought by TEACHTEXT increases. **(b) Performance vs teacher type.** We study the influence of teachers with different text embeddings at input: w2v and gpt2-xl. The first point represents the performance of the student without using TEACHTEXT. We observe a boost in performance independent of the nature of the teacher.

that there isn't much improvement that may be achieved by using a mixture of architectures as teachers. This is somehow expected, since these methods usually share the same video modalities so there isn't much additional information that may be captured by the combination of multiple architectures. However, we expect to get a further boost if we diversify the set of used modalities.

7.7. Architecture extension

In addition to the main paper, we also introduce a new CE-L base architecture. This is similar to the CE [21] and CE+, but uses `w2v` as the text embedding. In this way, the number of parameters are greatly reduced, making this the most lightweight architecture in terms of number of parameters that we can create. In Tab. 1, you can see that our method TEACHTEXT is effective even when using this lightweight architecture. This architecture also has the lowest numbers of parameters when compared to other state of the art methods as can be seen in Tab.2,3,4,5,6,7,8,9.

Model	MSRVTT		MSRVTT 1k-A		MSVD		DiDeMo		LSMDC		ActivityNet	
	Base	TEACHTEXT	Base	TEACHTEXT	Base	TEACHTEXT	Base	TEACHTEXT	Base	TEACHTEXT	Base	TEACHTEXT
MoEE	$24.4_{\pm 0.1}$	$\mathbf{25.8}_{\pm 0.1}$	$41.6_{\pm 0.4}$	$\mathbf{43.4}_{\pm 0.6}$	$41.8_{\pm 0.3}$	$\mathbf{43.2}_{\pm 0.5}$	$33.2_{\pm 1.4}$	$\mathbf{40.2}_{\pm 0.7}$	$23.8_{\pm 0.4}$	$\mathbf{26.0}_{\pm 0.5}$	$40.1_{\pm 0.3}$	$\mathbf{45.2}_{\pm 0.1}$
CE	$24.4_{\pm 0.1}$	$\mathbf{25.9}_{\pm 0.1}$	$42.0_{\pm 0.8}$	$\mathbf{43.8}_{\pm 0.3}$	$42.3_{\pm 0.6}$	$\mathbf{42.6}_{\pm 0.4}$	$34.2_{\pm 0.4}$	$\mathbf{39.5}_{\pm 0.5}$	$23.7_{\pm 0.3}$	$\mathbf{25.5}_{\pm 0.5}$	$40.4_{\pm 0.3}$	$\mathbf{45.0}_{\pm 0.6}$
MMT	-	-	$44.7_{\pm 0.4}$	$\mathbf{45.6}_{\pm 0.7}$	-	-	-	-	$24.6_{\pm 0.7}$	$\mathbf{25.9}_{\pm 0.6}$	$44.0_{\pm 0.4}$	$\mathbf{47.9}_{\pm 0.4}$
CE+	$29.2_{\pm 0.2}$	$\mathbf{30.4}_{\pm 0.0}$	$50.3_{\pm 0.2}$	$\mathbf{50.9}_{\pm 0.4}$	$46.5_{\pm 1.0}$	$\mathbf{46.6}_{\pm 0.5}$	$35.8_{\pm 0.4}$	$\mathbf{40.4}_{\pm 0.4}$	$28.1_{\pm 0.3}$	$\mathbf{30.7}_{\pm 0.3}$	$39.7_{\pm 0.0}$	$\mathbf{46.3}_{\pm 0.2}$
CE-L	$25.5_{\pm 0.1}$	$\mathbf{26.9}_{\pm 0.1}$	$45.7_{\pm 0.2}$	$\mathbf{46.5}_{\pm 0.8}$	$41.3_{\pm 0.5}$	$\mathbf{42.6}_{\pm 0.7}$	$36.4_{\pm 0.5}$	$\mathbf{41.5}_{\pm 0.4}$	$24.1_{\pm 0.2}$	$\mathbf{25.9}_{\pm 0.3}$	$39.6_{\pm 0.5}$	$\mathbf{45.7}_{\pm 0.2}$

Table 1. **Method generality**. Retrieval performance on various datasets when applying TEACHTEXT on top of different base models. In addition to the main paper, we present results on the CE-L architecture which has a significant drop in the number of used parameters as compared to the other models. We present in bold cases where TEACHTEXT brings an improvement over the base architecture. As can be seen, our method is effective and brings a consistent boost independent of the base architecture.

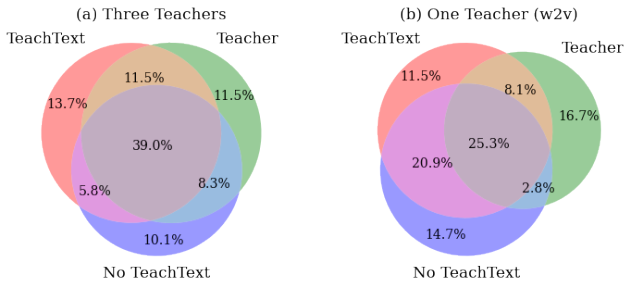

Figure 6. **Share of samples correctly retrieved samples in terms of R1 when using TEACHTEXT on the MSR-VTT test set.** In **sub-fig (a)** we show the case where we learn from 3 teachers, while in **sub-fig (b)** you can find the single teacher case. We can see that the model with TEACHTEXT, preserves most of the knowledge from the student without TEACHTEXT, but also acquires new information from the teacher (yellow area). Best viewed in color.

7.8. Model complexity

Changes in the pretrained text embedding strongly affect the number of parameters. Because of this factor, using more text embeddings at test time may strongly affect the total number of learnable parameters available to the model (in addition to adding the requirement to extract additional text embeddings during inference). While the simple 'Mean' aggregation from Fig.5b in the main paper, does not change the number of parameters, the 'Concat' aggregation adds a significant quantity (approx 240M learnable parameters, yielding total model sizes of 503.98M vs 262.73M for CE+). The proposed TEACHTEXT approach leaves the number of parameters untouched.

Since changing the text embedding to CE+ results in an increase in number of learnable parameters, we also study a CE-L architecture as a lightweight alternative in this Suppl. Mat. (described in Sec. 7.7), which demonstrates that the gain from the proposed TEACHTEXT approach is not limited to models with many learnable parameters. Please check Tab.2,3,4,5,6,7,8,9 for the exact number of params for every used architecture.

7.9. Amount of training data vs performance.

We next study how training data quantity influences the proposed method. In Fig. 5a we observe that by using the TEACHTEXT with more and more data, the performance gap increases, suggesting that its benefit may prove to be useful even in larger scale dataset scenarios.

7.10. Teacher study

In Fig. 5b, we study how each embedding affects the final performance. We observe that even though the model ingesting w2v embeddings has a significant lower performance than the student model without using TEACHTEXT, there is a significant gain when learning from the teacher which uses w2v. This again indicates that there is additional information captured by using a different text embedding which can be exploited by TEACHTEXT.

7.11. Influence of distillation over the correctly retrieved samples

In Fig. 6 we present the shares of correctly retrieved samples in terms of R1 on the test set of the MSR-VTT dataset for the student with and without TEACHTEXT and for the teacher. In Fig. 6a we present results when we learn from the three teachers and in Fig. 6b we considered the case when we learn only from one teacher (namely w2v). There is a significant share of correctly retrieved sample between the student using TEACHTEXT and the teacher.

8. Comparison to prior work

In Tab.2,3,4,5,6,7,8,9 we make an extensive comparison of our method with other methods from the literature. In addition to the numbers reported in the main paper, we also report results for the `v2t` task. Moreover, we present the number of parameters of each method where available. As can be seen, our TEACHTEXT algorithm brings a clear improvement and the total number of parameters remains the same as for the base architecture. In addition to the main paper, we also introduce a new CE-L base architecture. This is similar to the CE [21] and CE+, but uses `w2v` as text embedding. In this way, the number of parameters is greatly reduced. As can be seen from the tables, this lightweight architecture combined with our TEACHTEXT algorithm has very good results showcasing the effectiveness of TEACHTEXT across different parameter regimes. Moreover, some qualitative results can be seen in Fig.7.

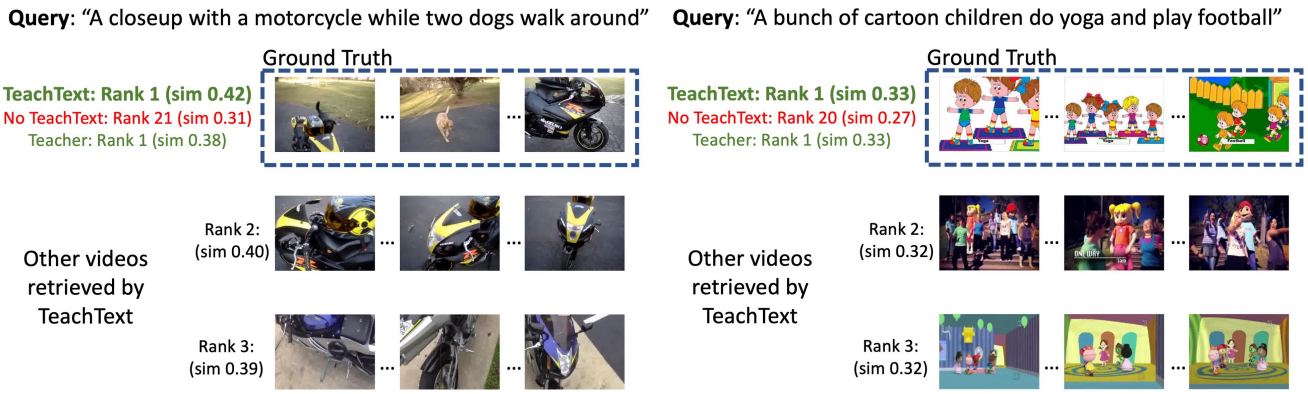

Figure 7. **Qualitative results.** We present the top 3 video retrievals for each query, given by the TEACHTEXT method used on top of a CE+ architecture. Moreover, we show the rank and similarity for the teacher, as well as for the student without using TEACHTEXT for the ground truth video. We mark in green cases where the retrieval is correct in terms of R1 and with red cases where is incorrect. For each of the cases shown, the model learns from the teacher to correct its prediction.

Model	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Params
Dual[9]	t2v	7.7	22.0	31.8	32.0	v2t	13.0	30.8	43.3	15.0	-
HGR[7]	t2v	9.2	26.2	36.5	24.0	v2t	15.0	36.7	48.8	11.0	-
MoEE[25]$^{??}$	t2v	$11.1_{\pm 0.1}$	$30.7_{\pm 0.1}$	$42.9_{\pm 0.1}$	$15.0_{\pm 0.0}$	v2t	$16.5_{\pm 0.1}$	$43.1_{\pm 0.5}$	$57.3_{\pm 0.6}$	$7.7_{\pm 0.5}$	$400.41M$
CE[21]$^{??}$	t2v	$11.0_{\pm 0.0}$	$30.8_{\pm 0.1}$	$43.3_{\pm 0.3}$	$15.0_{\pm 0.0}$	v2t	$17.0_{\pm 0.5}$	$43.5_{\pm 0.4}$	$57.8_{\pm 0.5}$	$7.2_{\pm 0.2}$	$183.45M$
TT-CE	t2v	$11.8_{\pm 0.1}$	$32.7_{\pm 0.1}$	$45.3_{\pm 0.1}$	$13.0_{\pm 0.0}$	v2t	$19.3_{\pm 0.4}$	$47.0_{\pm 0.7}$	$60.0_{\pm 0.4}$	$6.7_{\pm 0.5}$	$183.45M$
TT-CE-L	t2v	$13.0_{\pm 0.0}$	$34.6_{\pm 0.1}$	$47.3_{\pm 0.2}$	$12.0_{\pm 0.0}$	v2t	$22.4_{\pm 0.3}$	$50.4_{\pm 0.6}$	$63.8_{\pm 0.3}$	$5.3_{\pm 0.5}$	$66.72M$
TT-CE+	t2v	$\mathbf{15.0}_{\pm 0.1}$	$\mathbf{38.5}_{\pm 0.1}$	$\mathbf{51.7}_{\pm 0.1}$	$\mathbf{10.0}_{\pm 0.0}$	v2t	$\mathbf{25.3}_{\pm 0.1}$	$\mathbf{55.6}_{\pm 0.0}$	$\mathbf{68.6}_{\pm 0.4}$	$\mathbf{4.0}_{\pm 0.0}$	$262.73M$

Table 2. **MSR-VTT full split: Comparison to state of the art.**

Model	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Params
MoEE[25]$^{??}$	t2v	$21.6_{\pm 1.0}$	$50.8_{\pm 1.1}$	$65.6_{\pm 0.7}$	$5.3_{\pm 0.6}$	v2t	$22.4_{\pm 0.5}$	$51.2_{\pm 1.0}$	$66.1_{\pm 0.4}$	$5.2_{\pm 0.3}$	$400.41M$
CE[21]$^{??}$	t2v	$21.7_{\pm 1.3}$	$51.8_{\pm 0.5}$	$65.7_{\pm 0.6}$	$5.0_{\pm 0.0}$	v2t	$22.7_{\pm 0.4}$	$51.8_{\pm 0.4}$	$65.7_{\pm 0.2}$	$5.0_{\pm 0.0}$	$183.45M$
MMT[11]	t2v	$24.6_{\pm 0.4}$	$54.0_{\pm 0.2}$	$67.1_{\pm 0.5}$	$4.0_{\pm 0.0}$	v2t	$24.4_{\pm 0.5}$	$56.0_{\pm 0.9}$	$67.8_{\pm 0.3}$	$4.0_{\pm 0.0}$	$133.36M$
SSB[31]	t2v	27.4	56.3	67.7	**3.0**	v2t	26.6	55.1	67.5	**3.0**	–
TT-MMT	t2v	$24.8_{\pm 0.2}$	$55.9_{\pm 0.7}$	$68.5_{\pm 1.0}$	$4.3_{\pm 0.5}$	v2t	$25.1_{\pm 1.0}$	$57.1_{\pm 0.8}$	$69.9_{\pm 1.1}$	$4.0_{\pm 0.0}$	$133.36M$
TT-CE-L	t2v	$26.5_{\pm 0.4}$	$58.0_{\pm 0.8}$	$71.1_{\pm 0.4}$	$4.0_{\pm 0.0}$	v2t	$27.6_{\pm 0.8}$	$58.0_{\pm 0.6}$	$70.0_{\pm 0.5}$	$4.0_{\pm 0.0}$	$66.72M$
TT-CE+	t2v	$\mathbf{29.6}_{\pm 0.3}$	$\mathbf{61.6}_{\pm 0.5}$	$\mathbf{74.2}_{\pm 0.3}$	$\mathbf{3.0}_{\pm 0.0}$	v2t	$\mathbf{32.1}_{\pm 0.5}$	$\mathbf{62.7}_{\pm 0.5}$	$\mathbf{75.0}_{\pm 0.2}$	$\mathbf{3.0}_{\pm 0.0}$	$262.73M$

Table 3. **MSR-VTT 1k-A split[46]: Comparison with others.**

Model	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Params
VSE++[10]	t2v	15.4	39.6	53.0	9.0	v2t	21.2	43.4	52.2	9.0	-
M-Cues[27]	t2v	20.3	47.8	61.1	6.0	v2t	**31.5**	51.0	61.5	5.0	-
MoEE[25]$^{??}$	t2v	$21.1_{\pm 0.2}$	$52.0_{\pm 0.7}$	$66.7_{\pm 0.2}$	$5.0_{\pm 0.0}$	v2t	$27.3_{\pm 0.9}$	$55.1_{\pm 1.2}$	$65.0_{\pm 0.8}$	$4.3_{\pm 0.5}$	$131.37M$
CE[21]$^{??}$	t2v	$21.5_{\pm 0.5}$	$52.3_{\pm 0.8}$	$67.5_{\pm 0.7}$	$5.0_{\pm 0.0}$	v2t	$26.3_{\pm 1.4}$	$53.7_{\pm 0.4}$	$65.3_{\pm 1.1}$	$4.8_{\pm 0.2}$	$84.04M$
TT-CE	t2v	$22.1_{\pm 0.4}$	$52.2_{\pm 0.5}$	$67.2_{\pm 0.6}$	$5.0_{\pm 0.0}$	v2t	$26.0_{\pm 0.4}$	$53.3_{\pm 0.4}$	$63.9_{\pm 0.1}$	$4.9_{\pm 0.1}$	$84.04M$
TT-CE-L	t2v	$22.5_{\pm 0.0}$	$53.7_{\pm 0.3}$	$68.7_{\pm 0.5}$	$5.0_{\pm 0.0}$	v2t	$25.6_{\pm 0.2}$	$\mathbf{55.7}_{\pm 0.9}$	$65.9_{\pm 0.5}$	$\mathbf{4.0}_{\pm 0.0}$	$27.78M$
TT-CE+	t2v	$\mathbf{25.4}_{\pm 0.3}$	$\mathbf{56.9}_{\pm 0.4}$	$\mathbf{71.3}_{\pm 0.2}$	$\mathbf{4.0}_{\pm 0.0}$	v2t	$27.1_{\pm 0.4}$	$55.3_{\pm 1.0}$	$\mathbf{67.1}_{\pm 0.2}$	$\mathbf{4.0}_{\pm 0.0}$	$87.79M$

Table 4. **MSVD: Comparison to state of the art methods**.

Model	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$R@50\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$R@50\uparrow$	$MdR\downarrow$	Params
S2VT[41]	t2v	11.9	33.6	-	76.5	13.0	v2t	13.2	33.6	-	76.5	15.0	-
FSE[48]	t2v	$13.9_{\pm 0.7}$	$36.0_{\pm 0.8}$	-	$78.9_{\pm 1.6}$	$11.0_{\pm 0.0}$	v2t	$13.1_{\pm 0.5}$	$33.9_{\pm 0.4}$	-	$78.0_{\pm 0.8}$	$12.0_{\pm 0.0}$	-
MoEE[25]$^{??}$	t2v	$16.1_{\pm 1.0}$	$41.2_{\pm 1.6}$	$55.2_{\pm 1.6}$	$81.7_{\pm 1.4}$	$8.3_{\pm 0.5}$	v2t	$16.0_{\pm 1.5}$	$41.7_{\pm 1.9}$	$54.6_{\pm 1.7}$	$81.0_{\pm 1.4}$	$8.7_{\pm 0.9}$	$107.26M$
CE[21]$^{??}$	t2v	$17.1_{\pm 0.9}$	$41.9_{\pm 0.2}$	$56.0_{\pm 0.5}$	$83.4_{\pm 0.7}$	$8.0_{\pm 0.0}$	v2t	$17.1_{\pm 0.1}$	$41.8_{\pm 0.9}$	$55.2_{\pm 1.0}$	$83.0_{\pm 0.8}$	$7.7_{\pm 0.5}$	$79.29M$
TT-CE	t2v	$21.0_{\pm 0.6}$	$47.5_{\pm 0.9}$	$61.9_{\pm 0.5}$	$86.4_{\pm 0.8}$	$6.0_{\pm 0.0}$	v2t	$20.3_{\pm 0.6}$	$46.6_{\pm 0.6}$	$59.8_{\pm 1.2}$	$85.7_{\pm 0.6}$	$6.7_{\pm 0.5}$	$79.29M$
TT-CE-L	t2v	$\mathbf{22.3}_{\pm 0.2}$	$\mathbf{50.1}_{\pm 0.9}$	$\mathbf{64.3}_{\pm 0.5}$	$\mathbf{86.9}_{\pm 0.4}$	$\mathbf{5.3}_{\pm 0.5}$	v2t	$\mathbf{21.3}_{\pm 0.4}$	$\mathbf{48.3}_{\pm 0.5}$	$\mathbf{62.5}_{\pm 0.3}$	$86.6_{\pm 0.1}$	$\mathbf{6.0}_{\pm 0.0}$	$43.51M$
TT-CE+	t2v	$21.6_{\pm 0.7}$	$48.6_{\pm 0.4}$	$62.9_{\pm 0.6}$	$86.8_{\pm 0.3}$	$6.0_{\pm 0.0}$	v2t	$21.1_{\pm 0.2}$	$47.3_{\pm 0.2}$	$61.1_{\pm 0.4}$	$\mathbf{86.7}_{\pm 0.2}$	$6.3_{\pm 0.5}$	$99.51M$

Table 5. **DiDeMo: Comparison to state of the art methods**.

Model	Task	$R@1 \uparrow$	$R@5 \uparrow$	$R@10 \uparrow$	$MdR \downarrow$	Task	$R@1 \uparrow$	$R@5 \uparrow$	$R@10 \uparrow$	$MdR \downarrow$	Params
JSFus[46]	t2v	9.1	21.2	34.1	36.0	v2t	-	-	-	-	-
MoEE[25]$^{??}$	t2v	$12.1_{\pm 0.7}$	$29.4_{\pm 0.8}$	$37.7_{\pm 0.2}$	$23.2_{\pm 0.8}$	v2t	$11.9_{\pm 0.5}$	$28.0_{\pm 0.5}$	$37.4_{\pm 0.5}$	$25.5_{\pm 1.5}$	$159.78M$
CE[21]$^{??}$	t2v	$12.4_{\pm 0.7}$	$28.5_{\pm 0.8}$	$37.9_{\pm 0.6}$	$21.7_{\pm 0.6}$	v2t	$11.4_{\pm 0.4}$	$28.4_{\pm 0.7}$	$36.5_{\pm 0.5}$	$25.0_{\pm 0.8}$	$116.86M$
MMT[11]	t2v	$13.2_{\pm 0.4}$	$29.2_{\pm 0.8}$	$38.8_{\pm 0.9}$	$21.0_{\pm 1.4}$	v2t	$12.1_{\pm 0.1}$	$29.3_{\pm 1.1}$	$37.9_{\pm 1.1}$	$22.5_{\pm 0.4}$	$133.16M$
TT-MMT	t2v	$13.6_{\pm 0.5}$	$31.2_{\pm 0.4}$	$40.8_{\pm 0.5}$	$17.7_{\pm 0.5}$	v2t	$12.5_{\pm 0.3}$	$31.3_{\pm 0.6}$	$41.0_{\pm 1.1}$	$18.7_{\pm 0.5}$	$133.16M$
TT-CE-L	t2v	$14.2_{\pm 0.2}$	$30.6_{\pm 0.3}$	$40.0_{\pm 0.5}$	$20.3_{\pm 0.5}$	v2t	$13.6_{\pm 0.3}$	$30.8_{\pm 0.9}$	$38.9_{\pm 0.8}$	$21.5_{\pm 0.4}$	$87.22M$
TT-CE+	t2v	$\mathbf{17.2}_{\pm 0.4}$	$\mathbf{36.5}_{\pm 0.6}$	$\mathbf{46.3}_{\pm 0.3}$	$\mathbf{13.7}_{\pm 0.5}$	v2t	$\mathbf{17.5}_{\pm 0.6}$	$\mathbf{36.0}_{\pm 1.2}$	$\mathbf{45.0}_{\pm 0.5}$	$\mathbf{14.3}_{\pm 0.9}$	$388.24M$

Table 6. **LSMDC: Comparison to state of the art methods**.

Model	Task	$R@1 \uparrow$	$R@5 \uparrow$	$R@50 \uparrow$	$MdR \downarrow$	Task	$R@1 \uparrow$	$R@5 \uparrow$	$R@50 \uparrow$	$MdR \downarrow$	Params
MoEE[25]$^{??}$	t2v	$19.7_{\pm 0.3}$	$50.0_{\pm 0.5}$	$92.0_{\pm 0.2}$	$5.3_{\pm 0.5}$	v2t	$18.3_{\pm 0.5}$	$48.3_{\pm 0.8}$	$92.0_{\pm 0.2}$	$6.0_{\pm 0.0}$	$330.42M$
CE[21]$^{??}$	t2v	$19.9_{\pm 0.3}$	$50.1_{\pm 0.7}$	$92.2_{\pm 0.6}$	$5.3_{\pm 0.5}$	v2t	$18.6_{\pm 0.3}$	$48.6_{\pm 0.7}$	$92.0_{\pm 0.2}$	$6.0_{\pm 0.0}$	$260.68M$
HSE[47]	t2v	20.5	49.3	–	–	v2t	18.7	48.1	-	-	-
MMT[11]	t2v	$22.7_{\pm 0.2}$	$54.2_{\pm 1.0}$	$93.2_{\pm 0.4}$	$5.0_{\pm 0.0}$	v2t	$22.9_{\pm 0.8}$	$54.8_{\pm 0.4}$	$93.1_{\pm 0.2}$	$4.3_{\pm 0.5}$	$127.35M$
SSB[31]	t2v	**26.8**	58.1	93.5	**3.0**	v2t	**25.5**	57.3	93.5	**3.0**	-
TT-MMT	t2v	$25.0_{\pm 0.3}$	$\mathbf{58.7}_{\pm 0.4}$	$95.6_{\pm 0.2}$	$4.0_{\pm 0.0}$	v2t	$24.4_{\pm 0.1}$	$\mathbf{58.2}_{\pm 0.3}$	$95.7_{\pm 0.1}$	$4.0_{\pm 0.0}$	$127.35M$
TT-CE-L	t2v	$23.3_{\pm 0.1}$	$56.3_{\pm 0.1}$	$95.5_{\pm 0.1}$	$4.0_{\pm 0.0}$	v2t	$20.7_{\pm 0.2}$	$52.8_{\pm 0.2}$	$94.4_{\pm 0.0}$	$5.0_{\pm 0.0}$	$103M$
TT-CE+	t2v	$23.5_{\pm 0.2}$	$57.2_{\pm 0.5}$	$\mathbf{96.1}_{\pm 0.1}$	$4.0_{\pm 0.0}$	v2t	$23.0_{\pm 0.3}$	$56.1_{\pm 0.2}$	$\mathbf{95.8}_{\pm 0.0}$	$4.0_{\pm 0.0}$	$376.02M$

Table 7. **ActivityNet: Comparison to state of the art methods**.

Model	Task	$R@1 \uparrow$	$R@5 \uparrow$	$R@10 \uparrow$	$MdR \downarrow$	Task	$R@1 \uparrow$	$R@5 \uparrow$	$R@10 \uparrow$	$MdR \downarrow$	Params
VSE[18]	t2v	28.0	64.3	76.9	3.0	v2t	–	–	–	–	–
Dual[9]	t2v	31.1	67.4	78.9	3.0	v2t	–	–	–	–	–
VSE++[10]	t2v	33.7	70.1	81.0	2.0	v2t	–	–	–	–	–
HGR[7]	t2v	35.1	73.5	83.5	2.0	v2t	–	–	–	–	–
SSB[31]	t2v	44.6	81.8	89.5	**1.0**	v2t	58.1	83.8	90.9	**1.0**	–
CE[21]	t2v	$47.9_{\pm 0.1}$	$84.2_{\pm 0.1}$	$91.3_{\pm 0.1}$	$2.0_{\pm 0.0}$	v2t	$60.7_{\pm 1.0}$	$89.0_{\pm 0.4}$	$94.9_{\pm 0.2}$	$\mathbf{1.0}_{\pm 0.0}$	$115.56M$
TT-CE	t2v	$49.7_{\pm 0.1}$	$85.6_{\pm 0.1}$	$92.4_{\pm 0.1}$	$2.0_{\pm 0.0}$	v2t	$62.1_{\pm 0.2}$	$90.0_{\pm 0.1}$	$95.3_{\pm 0.1}$	$\mathbf{1.0}_{\pm 0.0}$	$115.56M$
TT-CE-L	t2v	$51.5_{\pm 0.1}$	$86.5_{\pm 0.1}$	$92.6_{\pm 0.1}$	$\mathbf{1.0}_{\pm 0.0}$	v2t	$\mathbf{65.0}_{\pm 0.5}$	$90.3_{\pm 0.5}$	$95.9_{\pm 0.2}$	$\mathbf{1.0}_{\pm 0.0}$	$55.07M$
TT-CE+	t2v	$\mathbf{53.2}_{\pm 0.2}$	$\mathbf{87.4}_{\pm 0.1}$	$\mathbf{93.3}_{\pm 0.0}$	$\mathbf{1.0}_{\pm 0.0}$	v2t	$64.7_{\pm 0.3}$	$\mathbf{91.5}_{\pm 0.3}$	$\mathbf{96.2}_{\pm 0.1}$	$\mathbf{1.0}_{\pm 0.0}$	$223.1M$

Table 8. **VaTeX: Comparison to state of the art methods**.

Model	Task	$R@1 \uparrow$	$R@5 \uparrow$	$R@10 \uparrow$	$MdR \downarrow$	Task	$R@1 \uparrow$	$R@5 \uparrow$	$R@10 \uparrow$	$MdR \downarrow$	Params
MoEE[25]	t2v	$11.6_{\pm 1.3}$	$30.2_{\pm 3.0}$	$43.2_{\pm 3.1}$	$14.2_{\pm 1.6}$	v2t	$13.0_{\pm 3.1}$	$30.9_{\pm 2.0}$	$43.0_{\pm 2.8}$	$14.5_{\pm 1.8}$	$57.75M$
CE[21]	t2v	$13.9_{\pm 0.8}$	$37.6_{\pm 1.2}$	$48.3_{\pm 1.4}$	$11.3_{\pm 0.6}$	v2t	$13.7_{\pm 0.7}$	$35.2_{\pm 2.7}$	$46.9_{\pm 3.2}$	$12.3_{\pm 1.5}$	$30.82M$
TT-CE	t2v	$14.2_{\pm 1.4}$	$36.6_{\pm 2.0}$	$\mathbf{51.1}_{\pm 2.1}$	$\mathbf{9.7}_{\pm 1.2}$	v2t	$14.1_{\pm 0.5}$	$34.8_{\pm 3.0}$	$\mathbf{49.1}_{\pm 0.3}$	$\mathbf{11.3}_{\pm 0.5}$	$30.82M$
TT-CE+	t2v	$\mathbf{14.4}_{\pm 0.5}$	$\mathbf{37.7}_{\pm 1.7}$	$50.9_{\pm 1.6}$	$9.8_{\pm 1.0}$	v2t	$\mathbf{14.3}_{\pm 0.6}$	$\mathbf{36.3}_{\pm 0.9}$	$48.3_{\pm 1.2}$	$\mathbf{11.3}_{\pm 0.5}$	$30.82M$

Table 9. **QuerYD: Comparison to state of the art methods**.

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017.

[2] Andrea Burns, Reuben Tan, Kate Saenko, Stan Sclaroff, and Bryan A Plummer. Language features matter: Effective language representations for vision-language tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7474–7483, 2019.

[3] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.

[4] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. VGGFace2: A dataset for recognising faces across pose and age. 2018.

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[6] David L Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics, 2011.

[7] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10638–10647, 2020.

[8] 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)*, pages 4171–4186, 2019.

[9] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, and Xun Wang. Dual dense encoding for zero-example video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] 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.

[11] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. *European Conference on Computer Vision*, 2020.

[12] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12046–12055, 2019.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, 2016.

[14] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing*. 2017.

[15] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

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

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

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[20] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[21] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*, 2019.

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

[23] Yang Liu, Zhaowen Wang, Hailin Jin, and Ian Wassell. Synthetically supervised feature learning for scene text recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 435–451, 2018.

[24] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.

[25] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018.

[26] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[27] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 19–27, 2018.

[28] Andreea-Maria Oncescu, Joao F. Henriques, Yang Liu, Andrew Zisserman Zisserman, and Samuel Albanie. Queryd: a video dataset with high-quality textual and audio narrations.

arXiv preprint arXiv:2011.11071, 2020.

[29] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[31] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, João Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. *arXiv preprint arXiv:2010.02824*, 2020.

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *preprint*, 1(8):9, 2019.

[34] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 123(1):94–120, 2017.

[35] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2017.

[36] FFmpeg team. Ffmpeg. `https://www.ffmpeg.org/`. Accessed: 2020-04-30.

[37] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[38] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[40] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[41] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[42] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 4581–4591, 2019.

[43] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[44] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[45] Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[46] 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)*, pages 471–487, 2018.

[47] Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 374–390, 2018.

[48] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems*, pages 9597–9608, 2019.

[49] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.