Everything at Once – Multi-modal Fusion Transformer for Video Retrieval Supplementary Material

Nina Shvetsova [1] Brian Chen [2] Andrew Rouditchenko[3] Samuel Thomas[4,5]
Brian Kingsbury[4,5] Rogerio Feris[4,5] David Harwath[6] James Glass[3] Hilde Kuehne[1,5]
[1]Goethe University Frankfurt, [2]Columbia University, [3]MIT CSAIL [4]IBM Research AI, [5]MIT-IBM Watson AI Lab, [6] UT Austin
shvetsov@uni-frankfurt.de

Supplementary material is organized as follows: first, we provide additional experimental results in Section A; then, we perform a qualitative analysis of zero-shot text-to-video retrieval in Section B; and finally, we provide more implementation details in Section C.

A. Additional Experimental Evaluation

A.1. CLIP Backbone

We additionally tested our model with stronger visual and text backbones. Namely, we used CLIP backbones (Contrastive Language-Image Pre-training) [7] pre-trained on the large Wikipedia-based image-text WiT dataset. We used the ViT-B/32 model and extracted one 512-dimensional feature per second for video and one 512-dimensional feature per word for text. For both modalities, we adopt features after projection into the multi-modal embedding space. Performance of zero-shot text-to-video retrieval and text-to-video retrieval after fine-tuning is presented in Table 1. We note that using CLIP features is especially beneficial for the MSR-VTT dataset, but performance on YouCook2 also improves compared to R152 + RX101 and word2vec backbones. We also note that performance on MSR-VTT after fine-tuning is coming close to the performance of the CLIP4CLIP [5] model that, however, is not directly comparable to ours. Compared to CLIP4CLIP, we are not fine-tuning backbones and we also are using a smaller MSR-VTT train subset for fine-tuning (7,000 clips compared to 9,000 clips).

Method	Visual Backbone	Text Backbone	FT	YouCook2		MSR-VTT	
				R@5↑	R@10↑	R@5↑	R@10↑
Ours	R152+RX101	word2vec		40.7	51.3	23.8	31.8
Ours	CLIP	word2vec		42.7	54.0	29.0	38.7
Ours	CLIP	CLIP		42.6	54.3	32.5	42.4
CLIP4CLIP	CLIP	CLIP		-	-	57.0	66.9
Ours	R152+RX101	word2vec	✓	59.1	70.9	52.1	63.7
Ours	CLIP	word2vec	✓	62.1	72.6	60.7	72.7
Ours	CLIP	CLIP	✓	62.1	72.9	62.7	75.0
CLIP4CLIP	CLIP	CLIP	✓	-	-	70.7	80.5

Table 1. Text-to-video retrieval on the YouCook2/MSR-VTT in zero-shot and fine-tune settings with CLIP backbones. As the video representation, we again use va – the fused video and audio modalities. FT: fine-tuning on downstream task. We include CLIP4CLIP [5] for completeness but do directly compare because of different pre-training and a different MSR-VTT train subset.

A.2. Action Segmentation

Following [2] we additionally report temporal action segmentation performance on the CrossTask and Mining YouTube datasets as proposed in [4]. We measured a frame-wise video segmentation performance given the order of actions in a video. Following inference procedure [4] we computed temporal alignment of video frames based on similarity matrix to text labels by a Viterbi-decoding. Before decoding, we transferred the similarity matrix to class probabilities by applying softmax with temperature 0.05 across all labels over all videos (as we did in NCE during training). Segmentation performance is measured by an intersection over union $IoU = \frac{G \cap D}{G \cup D}$ – the ratio between the intersection of ground truth action G and prediction D and the union of them – as well as an intersection over detection $IoD = \frac{G \cap D}{D}$.

Method	CrossTask			Mining YouTube		
	Recall↑	IOD↑	IOU↑	Recall↑	IOD↑	IOU↑
Mining YouTube [4]	-	-	-	-	19.2	9.8
MCN [2]	35.1	**33.6**	**22.2**	18.1	32.0	**23.1**
Ours	**39.3**	32.5	18.5	**19.4**	**32.7**	**23.1**

Table 2. Evaluation of zero-shot action segmentation on the CrossTask/Mining YouTube. We report results for "R152 + RX101" visual backbone (the same as used in MCN [2]).

In Table 2 we show IoU and IoD for temporal action segmentation with a recall for step action localization. We observe that our method shows a marginal boost in temporal action segmentation on the Mining YouTube dataset while it does not benefit on the CrossTask dataset. However, we note that the segmentation evaluation procedure relies on the given order of steps in a video, while in the CrossTask dataset about 30% steps are missed and step orders are not

Method	Make Kimchi Rice	Pickle Cucumber	Make Banana Ice Cream	Grill Steak	Jack Up Car	Make Jello Shots	Change Tire	Make Lemonade	Add Oil to Car	Make Latte	Build Shelves	Make Taco Salad	Make French Toast	Make Irish Coffee	Make Strawberry Cake	Make Pancakes	Make Meringue	Make Fish Curry	Average
Fully-supervised baseline [10]	19.1	25.3	38.0	37.5	25.7	28.2	**54.3**	25.8	18.3	31.2	**47.7**	12.0	39.5	23.4	30.9	41.1	53.4	17.3	31.6
CrossTask [10]	13.3	18.0	23.4	23.1	16.9	16.5	30.7	21.6	4.6	19.5	35.3	10.0	32.3	13.8	29.5	37.6	43.0	13.3	22.4
HT100M *et al.* [6]	**33.5**	27.1	36.6	37.9	24.1	35.6	32.7	35.1	**30.7**	28.5	43.2	19.8	34.7	33.6	40.4	41.6	41.9	27.4	33.6
MCN	25.5	31.1	39.7	32.7	35.4	**36.8**	29.0	40.0	28.4	**33.8**	45.7	27.5	36.1	34.9	39.6	42.6	43.0	29.1	35.1
Ours	30.5	**41.2**	**46.5**	**46.6**	**38.9**	32.0	19.5	**48.9**	25.8	33.6	44.7	**29.1**	**40.7**	**36.9**	**50.7**	**44.1**	**63.1**	**33.6**	**39.2**

Table 3. Step action localization performance on the CrossTask [10] dataset: recalls corresponding to every specific task.

#blocks	#heads	hidden s.	batch size	YouCook2 R@5↑	YouCook2 R@10↑	MSR-VTT R@5↑	MSR-VTT R@10↑
1	64	4096	224×10	**40.7**	**51.3**	**23.8**	31.8
2	64	4096	224×5	37.3	47.6	23.2	**32.5**
2	64	4096	112×10	38.6	49.8	20.8	28.6
2	32	2048	224×10	38.1	49.1	22.6	30.9
4	16	1024	224×10	35.4	46.8	23.7	31.7

Table 4. Evaluation of different fusion transformer architectures. *#blocks* stands for a number of transformer blocks, *#heads* – for a count of attention heads; *hidden s.* denotes a hidden size of the transformer layers (that linearly depends on the number of heads); *batch size* denotes a training batch size where $x \times y$ means that we use a batch of x videos and randomly sample y clips per video.

Configuration	YouCook2 R@5↑	YouCook2 R@10↑	MSR-VTT R@5↑	MSR-VTT R@10↑
aligned text-audio	37.8	47.2	14.0	20.0
disentangled text-audio	37.2	45.7	17.9	25.0
disentangled text-audio + loss weighting	**40.7**	**51.3**	**23.8**	**31.8**

Table 5. Evaluation of disentangling of audio and text while training on the HowTo100M dataset as well weighting components in the loss function. *Aligned text-audio* and *disentangled text-audio* were trained without loss weighting, *disentangled text-audio + loss weighting* – with $\lambda_{t\_v} = 1$, $\lambda_{v\_a} = \lambda_{t\_a} = \lambda_{t\_va} = \lambda_{v\_ta} = \lambda_{a\_tv} = 0.1$ as proposed.

always correct [10], so we consider the step action localization recall as a primary metric for this dataset, where our method improves performance by 4% with respect to MCN [2] baseline.

A.3. CrossTask Specific Results

To further analyze step action localization performance, we considered recalls for every specific task of the CrossTask dataset in Table 3. We note that our method shows a significant boost in almost all cooking-related categories, like "Make Banana Ice Cream" or "Grill Steak" while does not improve performance in not-cooking categories "Change Tire," "All Oil to Car," and "Build a Shelves." The MCN method, which also utilizes audio channel, similarly demonstrate a lower performance in "Change Tire" and "All Oil to Car" tasks compared to video-text-only the CrossTask [10] and HT100M [6] baselines. We can assume that this happens due to the fewer car-related video clips in the HowTo100M dataset (7.8M) compared to food-related clips (54.4 M).

A.4. Fusion Transformer Ablation

We also additionally ablate our Fusion Transformer with respect to the number of transformer blocks and the number of heads of multi-headed attention (and the hidden size of the transformer layer) in Table 4. Due to resource constraints, for an increase in the number of transformer blocks, we should linearly decrease either the number of heads or the training batch size (the large batch size is essential due to contrastive training). We observe that the best performing configuration consists of 1 transformer block and a maximum number of transformer heads (64 heads) that fits into resources, however, we assume the model can further boost performance by increasing the number of transformer blocks leveraging more resources.

A.5. Text-Audio Disentangling and Loss Weighting

We also show the importance of disentangling audio with respect to text while training on the HowTo100M dataset, as well as the importance of using a larger text-video loss weight in the loss function in Table 5. Since text is obtained by applying an ASR system to the audio track, to avoid text being learned just as an audio narration, we shift the audio clip randomly by half of clip length (4 seconds out of 8 seconds) with respect to the video and text boundaries in *"disentangled text-audio."* To further regularize text-audio learning, in *"+ loss weighting"* we used a larger weight for a text-visual loss $\lambda_{t\_v} = 1$ compared to other loss components $\lambda_{v\_a} = \lambda_{t\_a} = \lambda_{t\_va} = \lambda_{v\_ta} = \lambda_{a\_tv} = 0.1$ (similarly to [1]). Table 5 shows that both adaptations are beneficial for training on the HowTo100M dataset.

A.6. Relative Positional Encodings

As demonstrated in the paper, we found that absolute positional embeddings are not beneficial for our model. Apart

(a) Examples of clips retrieved in the top-1 results (@$R = 1$)

(b) Examples of clips retrieved in the top-5 results (@$R <= 5$)

(c) Examples of clips not retrieved in the top-5 results (@$R > 5$).

Figure 1. Qualitative evaluation. Examples of zero-shot text-to-video retrieval on the YouCook2. Each row shows the top-5 retrieved videos for a given text query. The correct video is highlighted with a red color.

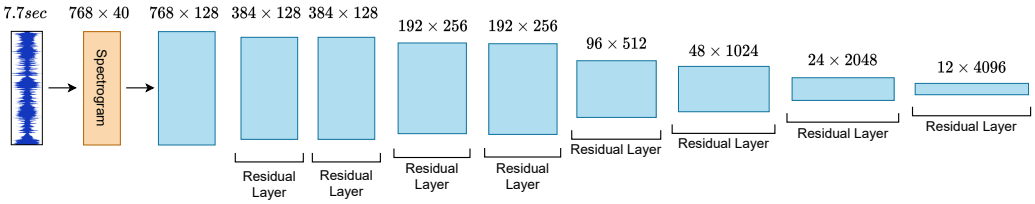
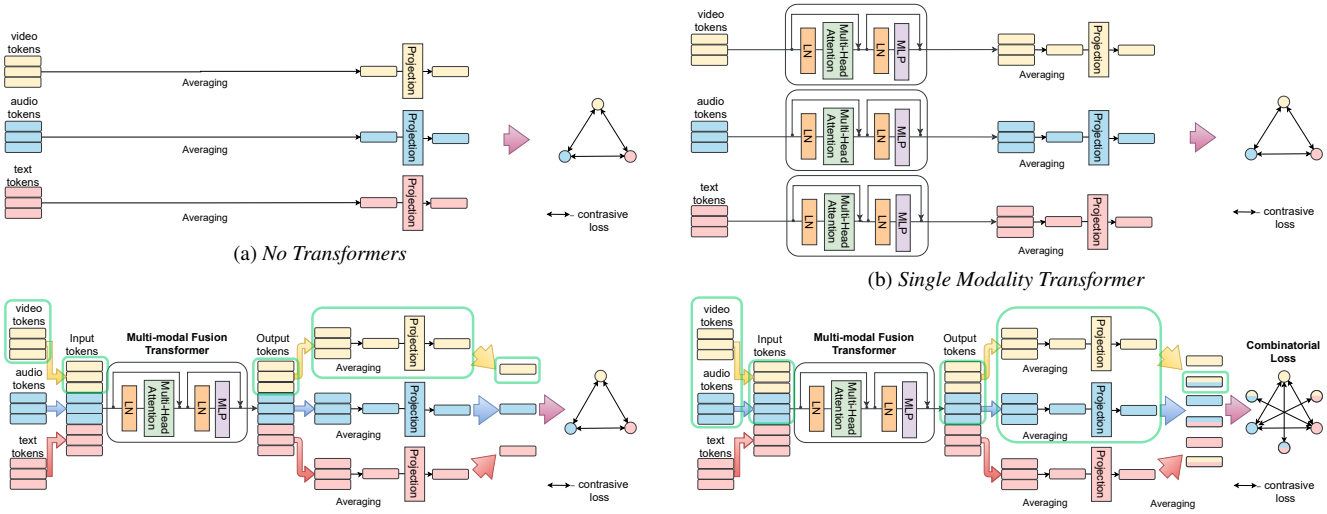


Figure 2. The schematic visualization of audio backbone network (the illustration is inspired by [8]).



(a) *No Transformers*

(b) *Single Modality Transformer*

(c) *Fusion Transformer*

(d) *Fusion Transformer + Combinatorial Loss*

Figure 3. Comparison of different architectures considered in ablation studies. Note that in illustration of the fusion transformer in (c) and (d), not all blocks are always active, using green rectangles we consider the video embedding computation in (c) and video-audio embedding computation in (d).

	YouCook2		MSR-VTT	
Configuration	R@5↑	R@10↑	R@5↑	R@10↑
RoPE [9]	40.4	51.2	23.1	31.6
no positional emb.	**40.7**	**51.3**	**23.8**	**31.8**

Table 6. Evaluation of Rotary Position Embedding (RoPE).

from them, we also tested relative positional encodings, namely the Rotary Position Embedding (RoPE) [9], that are shown to better generalize to longer inputs at test time. We incorporated RoPE into our attention block, where we independently apply RoPE to each sequence of text, video, or audio tokens. However, as presented in Table 6 we also found that RoPE does not benefit our model. But a more complex strategy that e.g., adds information about alignment tokens from different modalities (similarly to the RoPE 2D case) may lead to performance improvement.

B. Qualitative Analysis

We also qualitatively analyze the zero-shot retrieval capacity of our model on the YouCook2 dataset in Figure 1. We observe, that for all shown examples retrieved clips are semantically related to the given text query. Even when a correct video does not occur in the top-5 retrieval results, top-5 videos correspond to the text input: for example, for a query "bring a large pan of water to boil" our model predicts videos with boiling water in a pot.

C. Implementation Details

C.1. Audio Backbone

Following [2, 8], as an audio backbone, we use a trainable CNN with residual layers adopted from [3] that takes log-mel spectrograms with 16 kHz sampling rate, 25 ms Hamming window, 10 ms window stride, and 40 Mel filter bands. Note that this backbone is not pretrained. Since architecture used in [2, 8] extracts 6 1024-dimensional features per second, we adapt the last two residual blocks to extract ∼1.5 4096-dimensional features per second (the same as our video backbone). We illustrate architecture in Figure 2. While training on 8-seconds clips, we used 7.7 seconds of audio, that results exactly in 12 audio tokens.

C.2. Ablation Architectures

In Figure 3 we illustrate 4 architectures considered in our ablation studies: a) *no transformers*: our architecture without transformer layer, trained with three pairwise contrastive losses; 2) *single modality transformer*: leveraging three separate modality-specific transformers; 3) *fusion transformer*: the proposed modality agnostic transformer, but trained with three pairwise contrastive losses without fused modality components; 4) *fusion transformer + combinatorial loss*: the proposed architecture that utilises the modality agnostic transformer with combinatorial input, trained with combinatorial loss.

C.3. Fine-tuning Details

During fine-tuning on the YouCook2 and MST-VTT datasets, we set $\lambda_{t\_v} = \lambda_{v\_a} = \lambda_{t\_a} = \lambda_{t\_va} = \lambda_{v\_ta} = \lambda_{a\_tv} = 1$, and train the model for 5 epochs with a learning rate of $1e^{-5}$ and a batch size of 256 on the YouCook2 dataset, and for 25 epochs with the learning rate of $5e^{-5}$ and the batch size of 128 on the MSR-VTT dataset.

C.4. Training Time

Training our model on the HowTo100M dataset takes approximately 2 days on four Nvidia V100 32GB GPUs. Fine-tuning on the YouCook2 and the MSR-VTT takes less than 30 minutes.

References

[1] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *NeurIPS*, 2020. 2

[2] Brian Chen, Andrew Rouditchenko, Kevin Duarte, Hilde Kuehne, Samuel Thomas, Angie Boggust, Rameswar Panda, Brian Kingsbury, Rogerio Feris, David Harwath, et al. Multimodal clustering networks for self-supervised learning from unlabeled videos. *ICCV*, 2021. 1, 2, 4

[3] David Harwath, Adria Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *ECCV*, 2018. 4

[4] Hilde Kuehne, Ahsan Iqbal, Alexander Richard, and Juergen Gall. Mining youtube - a dataset for learning fine-grained action concepts from webly supervised video data. *arXiv preprint arXiv:1906.01012*, 2019. 1

[5] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*, 2021. 1

[6] 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*, 2019. 2

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 1

[8] Andrew Rouditchenko, Angie Boggust, David Harwath, Brian Chen, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Hilde Kuehne, Rameswar Panda, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, and James Glass. Avlnet: Learning audio-visual language representations from instructional videos. In *Interspeech*, 2021. 4

[9] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 4

[10] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019. 2