VALUE: A Multi-Task Benchmark for Video-and-Language Understanding Evaluation Supplementary Material

Linjie Li[*1], **Jie Lei**[*2], **Zhe Gan**[1], **Licheng Yu**[2], **Yen-Chun Chen**[1],
Rohit Pillai[1], **Yu Cheng**[1], **Luowei Zhou**[1], **Xin Eric Wang**[3], **William Yang Wang**[4],
Tamara L. Berg[2], **Mohit Bansal**[2], **Jingjing Liu**[5], **Lijuan Wang**[1], **Zicheng Liu**[1]
[1]Microsoft Corporation [2]UNC Chapel Hill
[3]UC Santa Cruz [4]UC Santa Barbara [5]Tsinghua University
{lindsey.li,zhe.gan,yen-chun.chen,rohit.pillai,
yu.cheng,luowei.zhou,lijuanw,zliu}@microsoft.com
{jielei,licheng,tlberg,mbansal}@cs.unc.edu
xwang366@ucsc.edu, william@cs.ucsb.edu, JJLiu@air.tsinghua.edu.cn

A Additional Data Statistics

We visualize video length distribution for each video data in Figure 1. Table 1 and 2 summarize the top-20 most frequent nouns and verbs in subtitles and annotations.

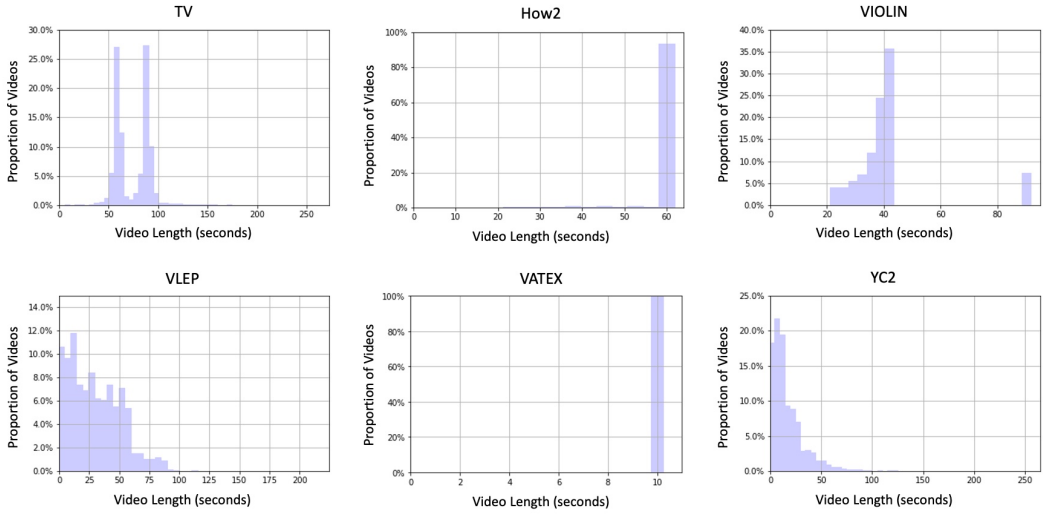

Figure 1: Visualization of video length distribution.

B Additional Results

B.1 Impact of Visual Representations

Table 3 shows the results using different visual representations. Our key observations are summarized as follows: (i) We confirm that image-text pre-trained CLIP-ViT features are generalizable to video-

[*] Equal contribution.

35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks.

Table 1: Top-20 most frequent nouns and verbs in subtitles. Character names have been filtered out from TV/VIOLIN/VLEP.

Dataset	Nouns	Verbs
TV	time, something, guy, way, right, man, thing, look, night, anything, god, someone, nothing, life, day, thank, kind, wait, woman, everything	know, go, think, want, see, need, tell, say, take, make, let, mean, come, find, give, look, talk, believe, love, feel
How2	bit, way, kind, time, water, today, thing, lot, side, video, music, right, let, something, oil, cup, sugar, top, part, half	make, want, go, see, know, use, take, put, need, add, let, show, think, start, give, look, keep, cut, come, say
VIOLIN	time, something, god, look, thing, way, right, man, guy, day, night, mom, anything, thank, kind, nothing, wait, life, baby, let	know, go, think, want, see, say, tell, take, need, let, come, make, mean, look, give, love, feel, talk, believe, call
VLEP	time, look, kind, something, thing, right, way, man, day, music, lot, everything, let, bit, guy, thank, food, night, place, god	know, go, think, see, want, need, make, take, let, say, come, love, look, mean, tell, give, try, feel, find, eat
VATEX	way, time, bit, side, look, god, job, alright, thing, kind, hand, man, today, video, day, right, lot, water, thank, something	go, know, see, want, make, take, think, let, put, say, come, need, keep, look, use, give, start, show, hold, love
YC2	bit, oil, water, salt, sauce, pepper, time, kind, teaspoon, heat, pan, cup, chicken, onion, half, way, butter, garlic, side, flavor	add, want, make, put, use, go, take, let, see, cook, know, need, give, mix, start, cut, keep, turn, think, look

and-language (VidL) tasks (L2 vs. L1/3/4). CLIP-ViT features lead to stronger performance than other 2D or 3D features. (ii) VALUE tasks also benefit from video-text pre-trained S3D features (L3 vs. L4). However, the performance improvement mostly comes from YC2 tasks, the videos of which are similar to the videos used to pre-train the S3D model. These results imply that the video domain of pre-training data is critical to downstream performance. (iii) When taking advantage of both 2D and 3D features, the model achieves the best performance (CLIP-ViT+SlowFast, L7), suggesting that both appearance and motion information are required to solve VALUE tasks. (iv) However, 2D and 3D features do not always complement each other. For example, performance on ResNet+S3D (L6) is worse than that on S3D alone (L4). (v) Retrieval and captioning tasks greatly depend on the quality of visual representations, while QA performance stays relatively stable with different features. This result agrees with our observations in Table 3 in the main text, where we show that QA tasks rely more on information from subtitle channel.

In addition, we finetune the pre-trained HERO on different visual representations (L9-12). Comparing their counterparts without pre-training, we observe a consistent performance improvement in almost all tasks across all visual representations examined. Note that in L10-12, though the video features used in pre-training is different from that used in finetuning, we still observe significant performance gains compared to models without pre-training. This suggests that the video-language alignment learned via HERO pre-training is transferrable to different visual representations.

B.2 Zero-Shot Evaluation on CLIP

Inspired by recent works [6, 13] leveraging image-text pre-training for video-and-language (VidL) tasks and the strong performance of CLIP-ViT features in Section B.1, we further perform a zero-shot evaluation using CLIP [18] on our retrieval and QA tasks. Captioning tasks cannot be directly evaluated as there is no decoder trained in CLIP.

CLIP is composed of an image encoder and a text encoder. To evaluate CLIP on VidL tasks, we first sample video frames from a video clip, then encode them via the image encoder from CLIP to obtain a sequence of frame features. For subtitle and textual query, we directly apply the text encoder from CLIP to generate a text representation.

First, we evaluate CLIP on the video-only input. For video retrieval tasks (*i.e.*, YC2R and VATEX-EN-R), we use mean pooling to aggregate the feature of all frames to obtain a global video representation.

Table 2: Top-20 most frequent nouns and verbs in annotations (query/question/caption). Character names have been filtered out from TVR/TVC/TVQA/VIOLIN/VLEP.

Dataset	Nouns	Verbs
TVR/TVC	walk, hand, room, door, table, conversation, patient, man, phone, woman, apartment, head, talk, bed, front, chair, coffee, arm, tell, couch	talking, walk, take, look, put, sitting, stand, turn, open, holding, sits, tell, hold, pick, asks, say, standing, give, go, looking
TVQA	room, patient, hand, door, color, apartment, table, phone, man, office, doctor, woman, shirt, something, couch, hospital, coffee, time, friend, guy	say, talking, said, tell, sitting, holding, asked, go, told, asks, going, walk, walked, want, wearing, talk, looking, come, give, standing
How2R	man, woman, video, lady, person, car, bowl, paper, hand, ingredient, girl, food, piece, plant, water, pan, glass, guy, kitchen, chef	make, explain, talking, show, using, showing, making, explaining, put, add, explains, cut, shown, cooking, describes, cutting, hold, holding, use, take
How2QA	video, color, man, woman, person, lady, hand, name, kind, type, bowl, food, ingredient, shirt, car, girl, item, table, boy, paper	used, shown, talking, using, put, wearing, added, make, holding, use, seen, cut, hold, want, mentioned, making, add, show, happen, explain
VIOLIN	man, woman, shirt, suit, hair, jacket, blonde, girl, lady, brunette, dress, boy, sweater, grey, friend, room, blue, men, pink, guy	wearing, tell, want, asks, sitting, say, explains, talking, trying, haired, see, go, think, make, walk, take, know, holding, going, look
VLEP	man, food, door, tell, room, woman, hand, patient, table, phone, shirt, walk, apartment, something, friend, vlogger, baby, girl, question, someone	say, tell, take, go, put, asks, look, give, start, walk, make, talk, going, leave, want, open, see, eat, turn, continue
VATEX-EN-R/-C	man, woman, person, people, boy, girl, group, hand, someone, music, child, ball, men, baby, water, room, piece, front, kid, floor	playing, using, sitting, play, holding, talking, standing, showing, make, wearing, shown, dancing, riding, put, show, stand, demonstrating, throw, demonstrates, hold
YC2R/YC2C	pan, oil, onion, salt, water, sauce, pepper, bowl, place, mix, pot, egg, potato, stir, chicken, mixture, slice, powder, butter, heat	add, cut, mix, put, cook, chopped, remove, fry, take, chop, cover, spread, serve, stir, pour, roll, drain, flip, blend, baking

Table 3: Impact of **visual representations**. ResNet(-152) [4] and SlowFast [3] are pre-trained on ImageNet [1] and Kinetics [5], respectively. S3D [21, 14] is pre-trained with video-text pairs in HowTo100M [15], OpenAI CLIP ViT [2, 18] is pre-trained with image-text pairs [18]. All results are reported on Val/Test (public) split. The best performance (of each block) are highlighted with bold (underline).

	Visual Feature	TVR	How2R	YC2R	VATEX-EN-R	TVQA	How2-QA	VIO-LIN	VLEP	TVC	YC2C	VATEX-EN-C	Meta-Ave
		AveR	AveR	AveR	AveR	Acc.	Acc.	Acc.	Acc.	C	C	C	
	2D Features, Finetune-only												
1	ResNet [4]	4.82	0.75	33.96	43.93	70.73	68.41	66.28	57.47	45.54	100.89	38.41	48.29
2	CLIP-ViT [2, 18]	7.93	1.52	35.93	62.87	71.07	69.34	66.80	58.27	48.99	112.25	52.42	53.40
	3D Features, Finetune-only												
3	SlowFast [3]	4.71	3.19	34.82	56.19	71.13	68.31	66.00	58.11	47.77	105.85	51.20	51.57
4	S3D [21, 14]	6.14	2.52	41.66	49.28	71.34	69.47	66.41	58.22	47.32	**125.58**	42.65	52.78
	2D+3D Features, Finetune-only												
5	ResNet+SlowFast	7.72	1.91	33.91	58.99	71.08	69.44	66.83	58.79	48.48	108.46	52.15	52.52
6	ResNet+S3D	5.16	2.32	33.88	46.19	70.70	66.68	68.60	58.65	45.22	105.83	39.51	49.34
7	CLIP-ViT+SlowFast	8.84	2.39	34.63	65.62	71.64	70.21	67.21	57.56	51.47	113.23	**56.97**	54.52
8	CLIP-ViT+S3D	6.66	2.27	36.68	62.35	70.27	68.54	67.06	59.13	50.05	110.18	52.77	53.27
	2D+3D Features, Pre-train on ResNet+SlowFast then Finetune												
9	ResNet+SlowFast	11.66	**5.97**	48.86	61.66	**74.80**	**74.32**	**68.98**	**67.40**	50.46	121.89	52.58	**58.05**
10	ResNet+S3D	9.45	5.20	47.81	47.00	72.65	72.68	67.71	65.94	46.42	117.11	38.77	53.70
11	CLIP-ViT+SlowFast	**12.92**	5.02	47.81	**66.49**	74.25	72.87	68.33	65.60	**51.56**	115.83	56.19	57.90
12	CLIP-ViT+S3D	11.85	5.43	**49.52**	63.37	72.56	73.64	67.92	65.82	50.26	117.58	50.73	57.15

Table 4: Zero-shot evaluation of CLIP [18] on retrieval and QA tasks. Results are reported on Val/Test (public) split. The best performance is highlighted in bold.

Input Channel	TVR AveR	How2R AveR	YC2R AveR	VATEX-EN-R AveR	TVQA Acc.	How2QA Acc.	VIOLIN Acc.	VLEP Acc.
Video-only	**0.13**	**0.0**	12.61	**55.68**	**27.00**	**54.54**	**52.40**	**55.35**
Video+Sub	**0.13**	**0.0**	**22.61**	46.78	23.17	44.94	50.00	56.35

Table 5: Additional results of **multi-task learning baselines** with **CLIP-ViT+SlowFast features** on Test (leaderboard) set. We compare the following model training settings: single-task training (ST), multi-task training (MT) by tasks or domains, all-task training (AT) and AT first then ST (AT → ST). The best performance (of each block) are highlighted with bold (underline).

	Training Setting	TVR AveR	How2R AveR	YC2R AveR	VATEX-EN-R AveR	TVQA Acc.	How2-QA Acc.	VIO-LIN Acc.	VLEP Acc.	TVC C	YC2C C	VATEX-EN-C C	Meta-Ave
1	Human	-	-	-	-	89.41	90.32	91.39	90.50	62.89	-	62.66	-
	Finetune-only												
2	ST	8.81	2.13	42.37	47.02	71.35	69.59	64.30	56.77	50.30	109.89	55.98	52.59
3	MT by Task	11.24	3.27	49.09	45.83	72.58	71.23	66.33	67.84	49.95	110.44	57.01	54.98
4	MT by Domain	11.30	2.66	46.24	44.69	73.66	71.20	66.59	68.13	49.52	104.39	56.25	54.06
5	AT	11.98	3.24	48.40	46.75	74.42	71.85	67.00	69.06	49.13	101.76	56.67	54.57
6	AT→ ST	12.40	3.61	50.93	**49.91**	74.38	71.88	66.80	68.68	49.41	110.63	**58.09**	56.07
	Pretrain on ResNet+Slowfast, then Finetune												
7	ST	**13.70**	3.38	56.59	46.66	74.52	73.82	64.19	67.10	**51.04**	120.22	55.30	56.96
8	MT by Task	13.45	**4.53**	**57.96**	47.47	73.56	73.95	65.80	68.32	49.30	121.66	55.10	57.37
9	MT by Domain	12.90	4.22	51.33	44.45	74.65	74.01	66.80	69.35	48.81	102.41	49.22	54.38
10	AT	12.55	3.32	52.16	46.58	**75.00**	73.69	**67.25**	68.65	48.81	114.27	54.79	56.10
11	AT→ ST	13.56	3.95	54.28	49.09	74.83	**74.60**	67.18	**69.37**	48.13	**121.89**	56.54	**57.58**

Cosine similarity is applied on the global video representation and the query representation to rank the relevance between video and query. For video corpus moment retrieval tasks (*i.e.*, TVR and How2R), an additional cosine similarity between each frame representation and query representation is used to predict the relevant span. Specifically, the localized span is determined by a sliding-window strategy. Similarly, we apply the same cosine similarity to QA tasks. For multiple-choice QA (*i.e.*, TVQA and How2QA), we concatenate the question with each answer choice as query, and calculate the similarity between the global video representation and the query representation. The answer with the highest similarity score among all answer choices is selected as the predicted answer. For VIOLIN and VLEP, which are formalized as binary classification problems, we generate the predictions according to a similarity score threshold. The best threshold is selected based on the validation set, and directly applied to the test set.

To augment the input with subtitle channel, we simply generate the subtitle sentence representations via text encoder and max pool them to aggregate the features of all subtitle sentences to obtain a global subtitle representation. Cosine similarity is applied to global subtitle representation and query representation to obtain a similarity score. The final similarity score is defined as the unweighted average of similarities scores generated from video-only input and subtitle-only input.

Results are reported in Table 4. Directly applying CLIP to YC2R and VATEX-EN-R achieves decent performance, which are consistent to observations in [13]. These results further support previous conclusions that image-text pre-training can benefit video-and-language tasks. However, on video moment retrieval tasks, where the model is required to localize the relevant moment based on the textual query, CLIP fails to differentiate among semantically similar video segments, resulting in poor performance. On QA tasks, where the queries or QA pairs are designed to be very similar to each other, CLIP without further finetuning struggles to predict the correct answer. Comparing video-only input to video+subtitle input, augmenting subtitle information does not guarantee performance improvement. The low performance may be due to ineffective video-subtitle fusion at prediction level or the limited capacity of CLIP to align subtitle information with textual query.

Table 6: Evaluation of **multi-task learning baselines** on Val/Test (public) split. Results are reported on HERO model with ResNet+SlowFast features unless specified otherwise. FT and PT denote finetuning and pre-training of the HERO model. We compare the following model training settings: single-task training (ST), multi-task training (MT) by tasks or domains, all-task training (AT) and AT first then ST (AT → ST). The best performance (of each block) are highlighted with bold (underline).

Training Setting	TVR	How2R	YC2R	VATEX-EN-R	TVQA	How2-QA	VIO-LIN	VLEP	TVC	YC2C	VATEX-EN-C	Meta-Ave
	AveR	AveR	AveR	AveR	Acc.	Acc.	Acc.	Acc.	C	C	C	
Finetune-only												
ST	7.72	1.91	33.91	58.99	71.08	69.44	66.83	58.79	48.48	108.46	52.15	52.52
MT by Task	7.23	2.70	39.03	57.64	71.23	71.65	66.82	66.64	47.24	111.35	51.07	53.87
MT by Domain	9.91	3.53	35.76	73.92	73.89	71.40	68.40	67.51	47.67	106.44	50.46	55.35
AT	9.93	3.29	38.58	72.84	74.36	71.85	67.62	66.99	46.73	100.00	51.07	54.96
AT → ST	10.53	4.42	41.18	74.06	73.89	71.56	68.97	66.37	47.59	108.30	51.87	56.26
Pre-train+Finetune												
ST	11.66	**5.97**	48.86	61.66	74.80	74.32	68.59	67.40	50.46	121.89	52.58	58.05
MT by Task	11.37	5.84	**49.27**	59.37	74.56	**74.86**	68.78	67.65	49.18	**130.38**	50.54	58.35
MT by Domain	10.97	4.56	42.18	75.44	74.79	75.15	68.60	68.26	47.88	109.30	45.96	56.65
AT	11.05	3.32	42.80	77.96	74.90	74.35	68.56	**69.24**	46.49	112.88	49.76	57.39
AT → ST	11.76	4.63	45.67	78.09	75.15	74.09	68.99	68.85	46.92	119.15	50.61	58.54
AT → ST on CLIP-ViT+SlowFast												
FT-only	12.34	5.12	42.46	78.72	75.33	73.19	69.05	67.99	**50.51**	114.60	**58.13**	58.86
PT+FT	**13.02**	5.66	45.33	**79.95**	**75.43**	74.57	**69.40**	69.19	49.67	115.65	56.35	**59.47**

B.3 Additional Results on Multi-Task Baselines

Table 5 presents results of proposed multi-task baselines with the optimal visual representations (CLIP-ViT+SlowFast) found in Section B.1. The highest meta-average score of 57.58 is achieved by AT → ST with pre-training (L11). A more concise version of the table is included in VALUE leaderbaord at https://value-benchmark.github.io/leaderboard.html.

Table 6 includes validation results of multi-task learning baselines. Table 7 presents more detailed results of multi-task learning baselines for retrieval and captioning tasks across different evaluation metrics on both validation split (Table 7a) and Test (leaderboard) split (Table 7b).

C Collection of Human Baselines

For multiple-choice QA tasks (*i.e.*, TVQA and How2QA), we resort to crowd-sourcing to obtain human baselines. Specifically, we present the human annotator with a multi-channel video, a question about the video, and a set of answer candidates. The annotator is asked to select the correct answer. Each question is presented to one annotator to evaluate human performance. For VIOLIN, a pair of video and hypothesis is presented to 3 human annotators, who are asked to determine whether the hypothesis is entailed or contradict to the video content. The human performance is evaluated based on the majority vote across the 3 human responses. For VLEP, human annotators are required to choose a more likely event from a pair of next event candidates based on the video content. We also take the majority vote to evaluate human performance. An example of our human evaluation interface is shown in Figure 2. The estimated hourly pay to our annotators is $8.6. The total amount spent is $2173.4.

For captioning tasks, we randomly sample one caption from the ground-truth annotations and use the rest as references to calculate the human performance across all captioning metrics. Note that in YC2C, there is only one caption collected for each video clip, thereby human performance is not reported.

D Additional Experimental Details

D.1 Downstream Adaptation

We describe in detail how HERO [10] architecture can be adapted to VALUE tasks.

For retrieval tasks, we add a query encoder head, consisting of a self-attention layer, two linear layers and an LN layer, on top of HERO's cross-modal transformer to obtain the query embeddings. The input multi-channal videos are encoded by cross-modal transformer and temporal transformer in HERO to obtain the contextualized video embeddings. For video moment retrieval tasks (TVR [9] and How2R [10]), we follow XML [9] to compute the matching scores between the query and visual frames both locally (frame-level, for moment retrieval) and globally (clip-level, for video retrieval). Specifically, we use cross-entropy loss to supervise the learning of the start and end index for local alignment and a combined hinge loss [22] over positive and negative query-video pairs for global alignment. For video retrieval tasks (YC2R [23] and VATEX-EN-R [20]), only the combined hinge loss is adopted.

For multiple-choice QA tasks (TVQA [7] and How2QA [10]), we append the corresponding QA pair (question and an answer candidate) to each of the subtitle sentences, which is fed into the cross-modal

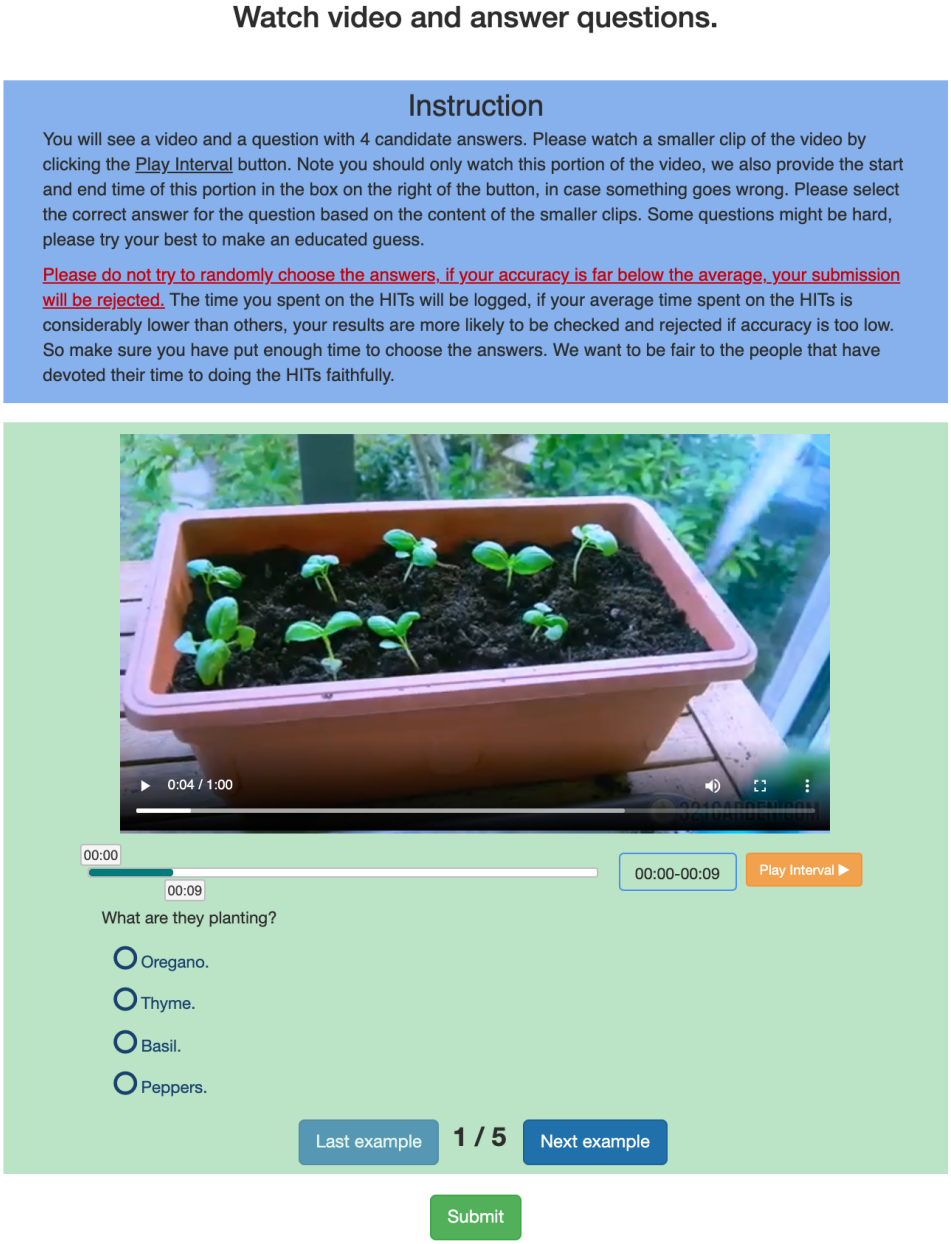

Figure 2: UI for huamn evaluation on video QA task.

transformer to perform early fusion with local textual context. In addition, these QA pairs are also appended to the input of temporal transformer to be fused with global video context. We use a simple attention layer to compute the weighted-sum-across-time of the QA-aware frame representations from the temporal transformer output. These final QA-aware global representations are then fed through an MLP and softmax layer to obtain the probability score of all the answers for the corresponding question. cross-entropy loss is used to supervise the model training. When supervision is available,[2] a span prediction loss (addition of two cross-entropy loss on start and end timestamps) is added as additional supervision.

Similar to multiple-choice QA, we append each natural language hypothesis in VIOLIN [11] (or next event candidate in VLEP [8]) to each of the subtitle sentences, as well as to the input of Temporal Transformer. A simple attention pooling layer is added to HERO to obtain the final query-aware global representations. We apply cross-entropy loss for the training.

For captioning tasks, a Transformer decoder [19] is employed to empower HERO with generative capabilities. We feed the whole subtitle-aligned video clip into HERO and obtain the subtitle-fused video representation for each frame. For TVC [9], frame representations are further grouped by the "moment of interest" using the time interval provided in the caption annotation, and the decoder-to-encoder attention is applied on the representations of the corresponding video moment. For YC2C [23] and VATEX-EN-C [20], as the caption is to describe the whole clip, the decoder-to-encoder attention is applied on the representations of the entire video. The decoder is trained with conventional left-to-right language modeling cross-entropy loss together with the HERO encoder end-to-end. We follow MMT [9] to use shallow Transformer decoder (2-layer) with 768-D hidden size, and simply use the greedy decoding at inference for constructing the baselines.

D.2 Video-Subtitle Fusion Methods

We introduce three early fusion baselines in detail. Let's denote the video segments embeddings as $\mathbf{F_V} = \{f_v, v \in \mathbf{V}\}$ and subtitle sentence embeddings as $\mathbf{F_S} = \{f_s, s \in \mathbf{S}\}$. The video segments embeddings are the concatenations of pre-extracted 2D appearance features concatenated with 3D motion features for each video segment. The subtitle sentence embeddings are obtained by max-pooling the contextualized subtitle token embeddings from a multi-layer transformer for each subtitle sentence. The first method (*sequence concat*) concatenates embeddings at sequence level without temporal alignment, denoted as $\mathbf{F_V}|\mathbf{F_S}$. The second method (*temporal align + sum*) takes the summation of the temporally aligned visual frame embeddings and subtitle sentence embeddings, denoted as $f_v + f_s$. The third method (*temporal align + concat*) concatenates the temporally aligned visual frame embeddings with subtitle sentence embeddings at feature level, denoted as $f_v|f_s$. Compared to HERO, we simply replace cross-modal transformer with methods described above. The fused embeddings from all the methods above are then fed into the temporal transformer to learn the global video context and obtain the final video embeddings.

D.3 Multi-Task Baselines

All our multi-task models are trained with a shared HERO encoder. We add only one head for each task type. For example, the same Transformer decoder is shared among different captioning tasks.

D.4 Implementation Details

Our models are implemented based on PyTorch [17].[3] To speed up training, we use Nvidia Apex[4] for mixed precision training. Gradient accumulation [16] is applied to reduce multi-GPU communication overheads. All experiments are run on 4 or 8 Nvidia V100 GPUs (32GB VRAM; NVLink connection) on Microsoft Azure.[5] We use AadmW [12] to optimize model parameters, with an initial learning rate in $\{3e-5, 5e-5, 1e-4\}$, β_1=0.9, β_2=0.98, and use learning rate warmup over the first 10% training steps followed by linear decay to 0.

[2] For example, TVQA and How2QA provides start and end timestamps to localize 'frames of interest' for the question.

[3] https://pytorch.org/

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

[5] https://azure.microsoft.com/

For single-task training, since the considered datasets vary in scale and domain, we use task-specific learning rates and training steps based on validation performance for each dataset. For multi-task training, we sample one task per mini-batch to train with a probability approximately proportional to the number of training examples for each task. The best checkpoint is selected based on the highest meta-average score achieved on validation split. To reproduce our results, please check the released starter code at https://github.com/VALUE-Leaderboard/StarterCode.

For YC2 and VATEX datasets, we employ ASR tool from Azure Cognitive Service[6] to generate the subtitles.

E License and Usage

As per the original authors, the annotations for TVQA [7], TVR [9], TVC [9], VIOLIN [11], YouCookII [23], VLEP [8], How2QA [10], How2R [10] are under CC BY-NC-SA 4.0 license[7], the annotations for VATEX [20] are under CC BY 4.0 [8]. The videos used in the datasets are from TV shows and YouTube, on non-offensive topics such as sitcoms and instructional videos. The annotations in the datasets do not contain personally identifiable information. Our released features are under CC BY-NC-SA 4.0 license[9], and our code is under MIT license[10].

The datasets used in the benchmark contain biases, both in the videos and the annotations. Such biases might be reflected in the predictions of the systems trained on these data. Users should not completely rely on such systems for making real-world decisions.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 3

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 3

[3] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[5] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 3

[6] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learningvia sparse sampling. In *CVPR*, 2021. 2

[7] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. Tvqa: Localized, compositional video question answering. In *EMNLP*, 2018. 6, 8

[8] Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. What is more likely to happen next? video-and-language future event prediction. In *EMNLP*, 2020. 7, 8

[9] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*, 2020. 6, 7, 8

[10] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *EMNLP*, 2020. 5, 6, 8

[11] Jingzhou Liu, Wenhu Chen, Yu Cheng, Zhe Gan, Licheng Yu, Yiming Yang, and Jingjing Liu. Violin: A large-scale dataset for video-and-language inference. In *CVPR*, 2020. 7, 8

[12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 7

[13] 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. 2, 4

[6] https://azure.microsoft.com/en-us/services/cognitive-services/speech-services/

[7] https://creativecommons.org/licenses/by-nc-sa/4.0/

[8] https://creativecommons.org/licenses/by/4.0/

[9] https://creativecommons.org/licenses/by-nc-sa/4.0/

[10] https://opensource.org/licenses/MIT

[14] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 3

[15] 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. 3

[16] Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*, 2018. 7

[17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 7

[18] 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. 2, 3, 4

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 7

[20] 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 *ICCV*, 2019. 6, 7, 8

[21] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 3

[22] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018. 6

[23] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 6, 7, 8

Table 7: Detailed results of multi-task training baselines on (a) Validation (Val/Test (public)) split and (b) Test (leaderboard) split of retrieval and captioning tasks. FT and PT denote finetuning and pre-training of the HERO model. We compare the following model training settings: single-task training (ST), multi-task training (MT) by tasks or domains, all-task training (AT) and AT first then ST (AT → ST). The best performance (of each block) are highlighted with bold (underline).

(a) Results on Validation split.

Training	TVR			How2R			YC2R			VATEX-EN-R			TVC				YC2C				VATEX-EN-C			
Setting	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	B@4	R	M	C	B@4	R	M	C	B@4	R	M	C
Human	-	-	-	-	-	-	-	-	-	-	-	-	12.90	36.50	20.60	64.56	-	-	-	-	24.65	46.85	24.89	61.86
Finetune-only																								
ST	3.38	8.63	11.14	0.62	2.01	3.09	18.73	37.17	45.82	29.72	67.48	79.77	11.31	33.31	17.21	48.48	9.64	37.15	16.98	108.46	29.64	48.03	22.70	52.15
MT by Task	2.47	7.88	11.34	0.77	3.09	4.25	22.51	42.73	51.86	29.28	65.92	77.75	10.72	32.76	17.07	47.24	9.87	36.71	17.03	111.35	28.87	47.33	22.50	51.07
MT by Domain	4.11	10.82	14.79	1.70	3.86	5.02	18.19	39.64	49.46	45.50	84.17	92.08	10.99	33.20	17.14	47.67	9.45	36.45	16.53	106.44	29.67	47.75	22.32	50.46
AT	4.23	10.77	14.78	1.00	3.48	5.40	20.90	43.01	53.06	44.12	83.00	91.80	10.43	32.61	16.88	46.96	9.28	35.82	16.45	102.26	30.14	48.11	22.45	51.00
AT → ST	4.67	11.46	15.45	1.78	5.17	6.64	23.17	45.31	55.07	45.66	84.10	92.42	10.50	32.85	16.96	47.59	9.59	37.05	17.06	108.30	29.74	48.18	22.60	51.87
Pre-train+Finetune																								
ST	5.57	12.43	16.99	**3.01**	**6.33**	**8.57**	**31.30**	53.04	62.23	33.04	70.31	81.64	**12.25**	**34.10**	17.54	50.46	11.47	39.79	18.14	121.89	30.07	48.34	22.72	52.58
MT by Task	5.17	12.16	16.79	2.78	6.18	8.57	30.18	**54.13**	**63.49**	31.17	67.62	79.31	11.53	33.61	17.42	49.18	**12.40**	**40.41**	**18.81**	**130.38**	29.22	47.57	22.37	50.54
MT by Domain	4.61	11.82	16.48	2.78	4.63	6.27	22.34	46.65	57.56	48.07	85.27	92.99	11.79	33.75	17.15	47.88	10.09	38.27	17.40	109.30	28.98	48.01	21.86	45.96
AT	5.08	12.08	15.98	1.47	3.55	4.94	23.80	47.31	57.30	51.72	87.90	94.26	10.92	32.96	16.86	46.49	10.54	38.04	17.36	112.88	29.72	48.13	22.26	49.76
AT → ST	5.49	12.61	17.18	2.62	5.02	6.25	27.38	50.09	59.53	52.08	87.92	94.26	11.01	33.06	16.94	46.92	11.17	38.89	17.82	119.15	29.66	48.19	22.52	50.61
AT → ST on ViT+SlowFast																								
FT-only	5.67	13.44	17.91	2.78	5.33	7.26	24.71	46.31	56.36	53.70	88.20	94.26	11.68	33.91	**17.63**	**50.51**	10.35	37.93	17.46	114.60	**32.93**	**50.13**	**24.06**	**58.13**
PT+FT	**5.93**	**14.36**	**18.76**	**3.01**	6.18	7.80	26.23	49.71	60.05	**55.62**	**89.15**	**95.07**	11.52	33.86	17.44	49.67	10.62	38.59	17.70	115.65	32.69	50.02	23.85	56.35

(b) Results on Test (leaderboard) split.

Training	TVR			How2R			YC2R			VATEX-EN-R			TVC				YC2C				VATEX-EN-C			
Setting	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	B@4	R	M	C	B@4	R	M	C	B@4	R	M	C
Human	-	-	-	-	-	-	-	-	-	-	-	-	13.05	36.52	20.65	62.89	-	-	-	-	24.77	46.88	25.09	62.66
Finetune-only																								
ST	3.10	8.44	11.55	0.32	1.74	3.16	24.00	44.26	53.80	15.97	42.14	56.91	11.26	32.96	16.91	46.76	9.00	36.35	16.58	106.24	29.63	48.02	22.70	52.16
MT by Task	2.99	8.21	12.06	0.24	1.90	3.56	27.62	51.49	60.04	16.45	41.99	56.08	10.69	32.60	16.87	46.01	9.54	35.51	16.62	105.22	28.86	47.32	22.49	51.07
MT by Domain	4.35	10.79	14.88	1.18	3.01	3.87	24.13	50.25	59.35	14.80	39.53	53.96	11.05	32.96	16.96	46.53	8.61	35.20	16.24	100.74	29.67	47.74	22.31	50.46
AT	4.30	10.57	14.42	0.71	2.45	4.11	27.81	53.57	62.34	15.63	40.80	55.56	10.89	32.75	16.84	46.46	9.18	35.21	16.27	101.72	30.21	48.10	22.49	51.07
AT → ST	4.82	11.43	15.03	1.34	2.84	4.03	31.17	54.11	63.15	16.52	42.42	56.79	10.65	32.69	16.75	46.12	9.00	36.32	16.67	103.73	29.74	48.20	22.61	51.87
Pre-train+Finetune																								
ST	5.69	13.38	17.06	1.90	4.43	5.93	38.97	63.65	71.01	**18.33**	**44.77**	**58.78**	**12.00**	**33.94**	17.39	48.97	11.46	**39.67**	**18.17**	**127.94**	30.07	48.32	22.80	52.57
MT by Task	6.01	13.71	18.18	1.82	**5.06**	**7.11**	**39.46**	**63.90**	**74.25**	17.95	43.84	58.12	11.56	33.42	17.21	48.02	**11.70**	38.90	18.13	123.40	29.21	47.58	22.36	50.49
MT by Domain	5.37	12.46	16.76	2.21	3.95	5.93	30.42	57.54	68.45	15.73	40.60	54.58	11.93	33.61	17.04	47.23	9.11	36.60	16.87	100.29	28.99	47.98	21.85	45.95
AT	5.35	12.61	16.87	2.14	4.03	5.93	31.86	57.54	67.21	16.48	41.40	56.15	11.19	33.10	16.84	46.04	10.17	37.40	17.21	109.11	29.70	48.13	22.26	49.74
AT → ST	5.72	13.21	17.57	**2.29**	4.98	6.25	34.73	59.85	67.89	17.08	42.59	56.90	11.40	33.09	16.88	46.38	11.32	39.03	17.83	120.86	29.62	48.18	22.52	50.59
AT → ST on CLIP-ViT+SlowFast																								
FT-only	5.91	13.58	17.72	1.42	3.79	5.61	31.73	55.67	65.40	25.01	55.36	69.36	11.61	33.84	**17.49**	**49.41**	9.94	37.62	17.59	110.63	**32.89**	**50.01**	**24.04**	**58.09**
PT+FT	**6.39**	**14.75**	**19.54**	1.90	3.79	6.17	35.10	59.48	68.27	24.51	54.34	68.41	11.78	33.68	17.15	48.13	11.25	39.02	17.92	121.89	32.68	**50.01**	23.84	56.29