Supplementary Material: Advancing High-Resolution Video-Language Representation with Large-Scale Video Transcriptions

Hongwei Xue*, Tiankai Hang*, Yanhong Zeng*, Yuchong Sun*,
Bei Liu, Huan Yang, Jianlong Fu, Baining Guo
Microsoft Research Asia
{v-honxue,v-tiahang,t-yazen,v-yuchongsun,bei.liu,huayan,jianf,bainguo}@microsoft.com

A. Author Contribution

The first four authors contribute equal to this research project. Among them, Hongwei Xue is responsible for model design, implementation of pre-training model and downstream video QA tasks. Tiankai Hang helps the model design, environment building for distributed training, and apply the pre-trained model for downstream extreme text-guided super-resolution task. Yanhong Zeng is in charge of the text-to-visual generation part, including the creation of dataset (FDVD), design and implementation of text-to-visual editing. Yuchong Sun is responsible for collecting and processing HD-VILA-100M dataset, discussing model design and implementation of downstream video-text retrieval tasks. Bei Liu, Huan Yang and Jianlong Fu oversee the whole project, including dataset collection and processing, pre-training model and downstream tasks design. Baining Guo provides valuable suggestions in paper organization and writing.

B. Limitation and Social Impact

The proposed video-language dataset and pre-training model show the capacity and generalization of learned video-language representation which could benefit many applications of computer vision and natural language processing. Pre-training with large scale of data results in much computation resource. How to reduce the model size and computing effort becomes more essential for future research. In addition, the usage of user generated data might bring the risk of bias. We tackle this problem by balancing various video categories, yet the videos might contain biased content. Moreover, how to avoid malicious usage of visual generation technique for conscious attack is also critical. However, these concerns are general to the entire fields and are not amplified by this work.

*Equal contribution in alphabetical order. This work was performed when Hongwei Xue, Tiankai Hang, Yanhong Zeng and Yuchong Sun were visiting Microsoft Research Asia as research interns. Corresponding authors: Bei Liu, Huan Yang, Jianlong Fu.

C. HD-VILA-100M Dataset Details

C.1. Video Duration and Transcript Length

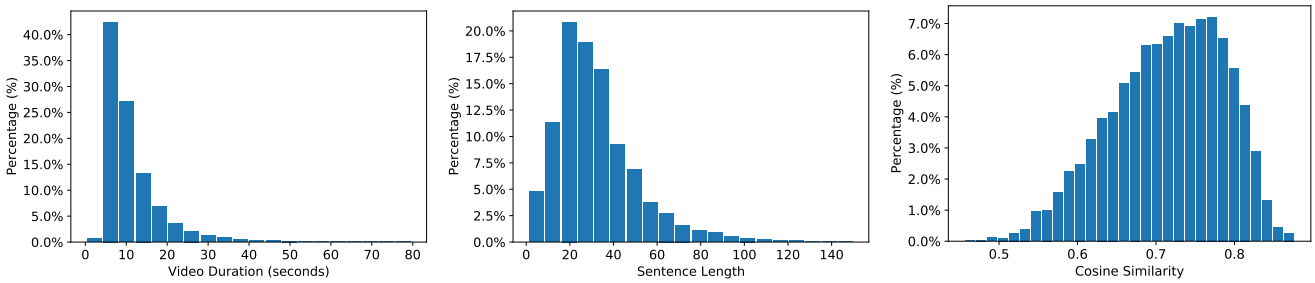
We plot the histogram of video clip duration and transcript length in Figure 1a and Figure 1b, respectively. From Figure 1a, we can see that most video clips in our dataset is between 5s to 15s, with an average of 13.4s. From Figure 1b, most sentences in HD-VILA-100M are between 15 words to 50 words, with an average of 32.5 words.

C.2. Semantic Richness

To analyze the semantic richness, we calculate the average unique n-grams and part-of-speech (POS) tags of transcriptions. We mainly compare them with HowTo100M [12] dataset as shown in Table 1. From the result, we can find that the sentences in our dataset have more n-grams and POS tags, which indicates more richness and diversity of semantics in our HD-VILA-100M dataset.

C.3. More Examples of HD-VILA-100M Dataset

Since we use transcripts as corresponding sentences for videos, the video-sentences are actually not all well aligned compared with video captioning datasets. Indeed, most of them are weakly related. We conduct an interesting experiment which uses our pre-trained model with the weakly aligned pairs to compute the similarity of these pairs. We show some examples with similarity scores in Figure 2. We can see that the pairs with higher score are well aligned. This indicates that, even with the weakly aligned video-transcript pairs, our pre-training model can learn a powerful embedding space between video and language. The similarity score distribution of video and text pairs in HD-VILA-100M is shown in Figure 1c.



(a) Distributions of video duration. (b) Distributions of sentence length. (c) Video-text similarity distribution.

Figure 1. More detailed statistics of HD-VILA-100M dataset.

Dataset	# avg unique n-grams			# avg POS tags			
	2-gram	3-gram	4-gram	noun	adj	adv	verb
HowTo100M [12]	1.77	2.08	1.46	2.25	0.85	0.68	0.20
HD-VILA-100M	4.18	13.08	20.89	6.63	1.88	2.07	5.09

Table 1. Statistics of average unique n-grams and POS tags. Our dataset has more unique n-grams and POS tags than HowTo100M [12]. The result indicates the transcriptions in HD-VILA-100M have richer and more diverse semantics.

D. Experiment Details

D.1. Video QA

MSRVTT-QA. MSRVTT-QA [20] is created based on video and captions in MSR-VTT [21], containing 10K videos and 243K open-ended questions. We follow the original work to use an answer vocabulary containing the most common 1.5K answers in the training and validation split as answer candidates. For each video, we randomly sample one segment for training and uniformly sample eight segments for testing. We resize HR frame of each segment to 720p and LR frames to 180p. In this task, we set #HR as 1 and #LR as 6. We use AdamW for optimization, with an initial learning rate of 1e-5, weight decay of 0.3, and set learning rate warm-up over the first 10% training steps followed by linear decay to 0. To alleviate over-fitting, we set dropout of Transformers to 0.1. We fine-tune our model on 8 NVIDIA Tesla V100 GPUs for 20 epochs with a batch size of 512. Gradient accumulation is applied to reach this batch size.

MSRVTT Multiple-Choice. MSRVTT multiple-choice test [22] is a multiple-choice task with videos as queries, and captions as answers. Each video contains five candidate captions, with only one positive match. The benchmark has 2,990 questions for the multiple-choice test. We directly inference our model trained on MSRVTT-Retrieval dataset to find the most positive match.

TGIF-QA. TGIF-QA [3] contains 165K QA pairs on 72K GIF videos. We experiment with three TGIF-QA tasks: *Action*, *Transition* and *FrameQA*. We randomly sample one segment for training and uniformly sample eight segments for testing. Each segment contains 1 HR frame and 6 LR frames for *Action* and *Transition*, 10 LR frames for *FrameQA*. Other settings are listed in Table 2. We use AdamW for optimization, and We fine-tune our model on 8 V100 GPUs, Gradient accumulation is applied to reach batch sizes listed in Table 2.

	Action	*Transition*	*FrameQA*
Epoch	80	80	40
Batch Size	384	384	448
Learning Rate	5e-5	5e-5	4e-5
Weight Decay	0.05	0.05	0.3
Drop Out	0.1	0.3	0.1

Table 2. Details of training Video QA on TGIF dataset.

D.2. Text-to-Video Retrieval

Due to the various resolution for videos in downstream datasets, we resize HR frame of each segment to 720p and LR frames to 180p. We adopt stage one model and the same training methods and objective for fine-tuning. We set the temperature to 0.08. We use learning rate warmup followed by multi-step learning rate decay. We adjust the number of sampled segments and frames according to the average time of videos for each dataset to cover about half of the video.

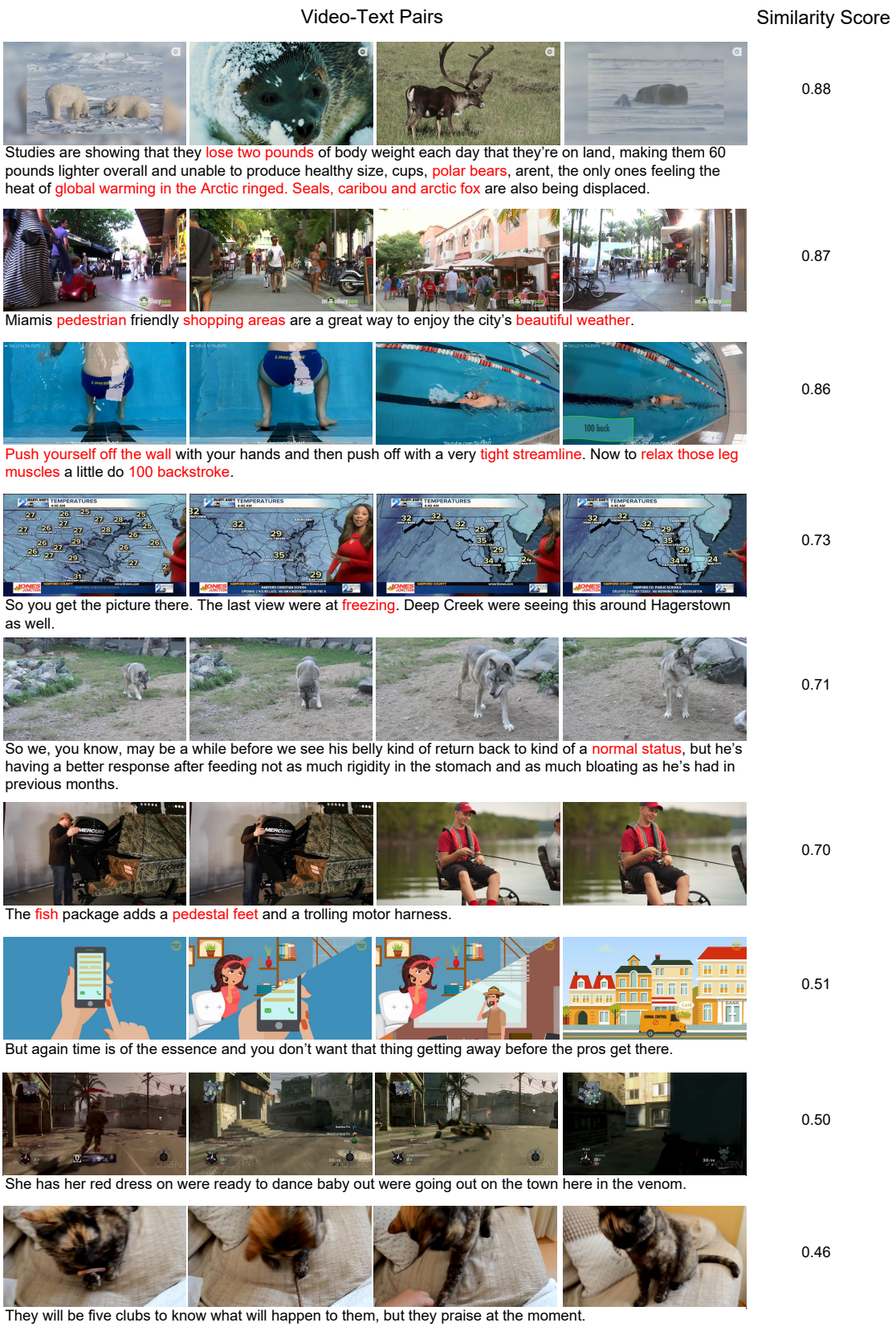


Figure 2. More examples of HD-VILA-100M with similarity scores calculated by HD-VILA. Relevant words are highlighted in red. [Best viewed in color.]

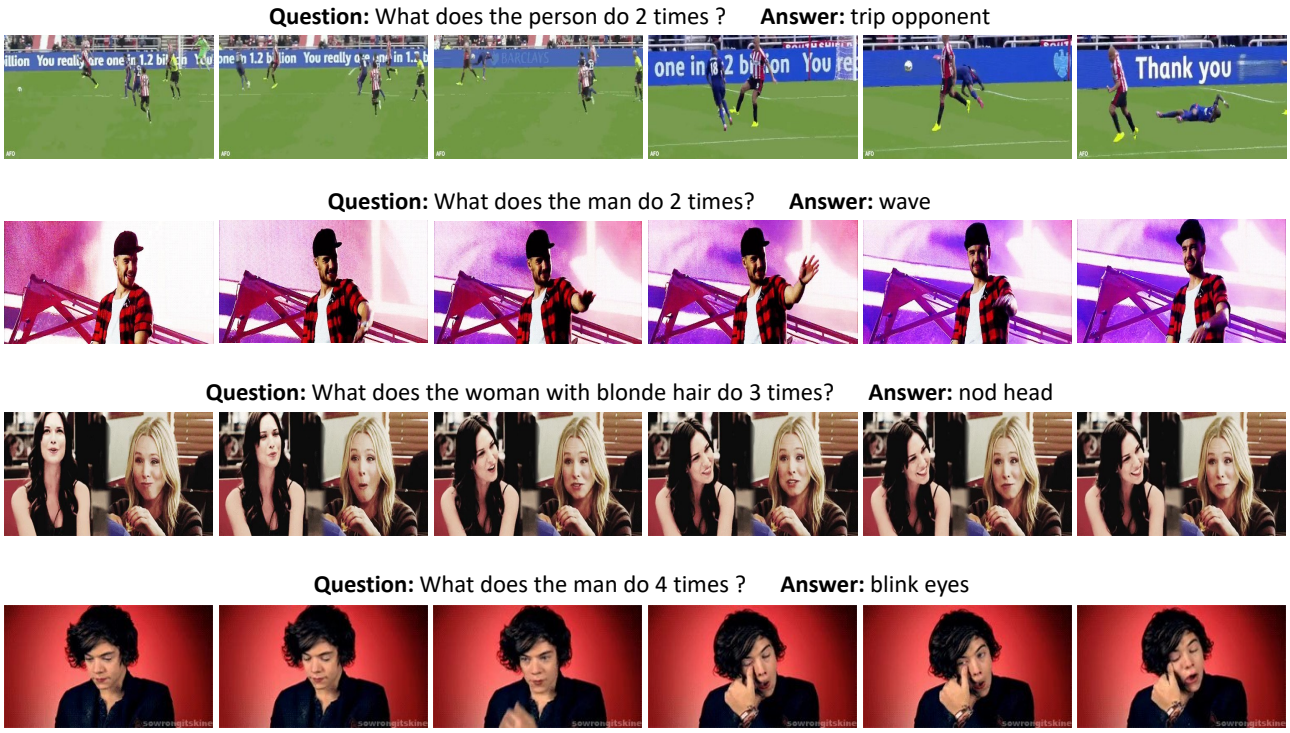


Figure 3. **Some examples for video QA task**. We take TGIF *Action* for example to demonstrate our model's ability to learn temporal information from videos.

For evaluation, we double the number of segments. More details for each tsak are given below

MSR-VTT. MSR-VTT [21] contains 10K YouTube videos with 200K descriptions. We follow previous works [8, 22], training models on 9K videos, and reporting results on the 1K-A test set. For zero-shot evaluation on low-resolution MSR-VTT videos, we uniformly sample 4 segments each with 11 frames. We crop a 224×320 patch for each frame and up-sample the middle frames by 4 times. In this setting, the sampled segments can nearly cover the videos on average. We remove the stop words in the text as [11]. We report the result of the last saved model of HD-VILA. When finetuning, we sample 2 segments for training and 4 segments for testing and each segment contains 11 frames. We use AdamW optimizer with an initial learning rate of 1e-5. We fine-tune the pre-trained model with 32 V 100 GPUs and the total batch size is 256.

DiDeMo. DiDeMo [1] consists of 10K Flickr videos annotated with 40K sentences. We follow [8, 23] to evaluate paragraph-to-video retrieval, where all descriptions for a video are concatenated to form a single query. When finetuning, we sample 4 segments for training and 8 segments for testing and each segment contains 11 frames. We use AdamW optimizer with an initial learning rate of 5e-6. We fine-tune the pre-trained model with 16 V 100 GPUs and the total batch size is 64.

LSMDC. LSMDC [15] consists of 118,081 video clips sourced from 202 movies. Each video has a caption. Evaluation is conducted on a test set of 1,000 videos. When finetuning, we sample 2 segments for training and 4 segments for testing and each segment contains 11 frames. We use AdamW optimizer with an initial learning rate of 5e-6. We fine-tune the pre-trained model with 8 V 100 GPUs and the total batch size is 64.

ActivityNet. ActivityNet Captions [7] contains 20K YouTube videos annotated with 100K sentences. We follow the paragraph-to-video retrieval protocols [8, 23] training on 10K videos and reporting results on the val1 set with 4.9K videos. When finetuning, we sample 4 segments for training and 8 segments for testing and each segment contains 13 frames. We use AdamW optimizer with an initial learning rate of 5e-6. We fine-tune the pre-trained model with 16 V 100 GPUs and the total batch size is 64.

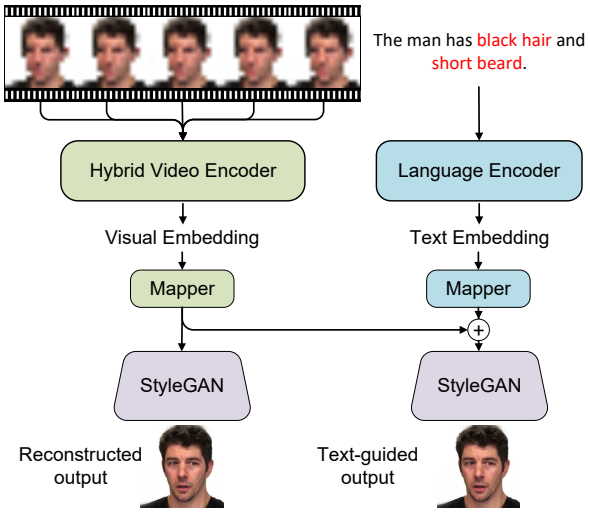


Figure 4. **Overview of our text-guided generation framework.** The framework consists of 1) two multi-modal encoders, 2) two mapper modules, and 3) a pre-trained StyleGAN [5]. First, the multi-modal encoders encode a video clip and a sentence to a visual and a text embedding, respectively. Second, the mapper modules map the embedding to the latent codes of StyleGAN. Finally, StyleGAN maps the latent codes w/ and w/o text information to images. See more details in Section D.3.1.

D.3. Text-to-Visual Generation

In this section, we introduce more details about text-to-visual generation tasks as a supplement to Section 5.4 in the main paper. We introduce the details of model design in Section D.3.1 and optimization objectives in Section D.3.2, following the introduction of our collected dataset of video-description pairs of the human faces in Section D.3.3. We provide more generation results and experimental analysis in Section D.3.4.

D.3.1 Model Design

To achieve the text-to-visual generation tasks by our pre-trained model HD-VILA, we follow previous works to combine the cross-modality encoders of HD-VILA and a well pre-trained generation model, StyleGAN [5], in our framework [13, 19]. The overview of our text-to-visual generation framework is shown in Figure 4. Specifically, our framework consists of three key components, including 1) two multi-modal (visual/text) encoders, 2) two visual/text mapper modules, and 3) a pre-trained StyleGAN. We introduce more details of each component as below.

Multi-Modal Encoders To deal with multi-modal inputs, we inherit the hybrid video encoder and the language encoder from our pre-trained model HD-VILA. Specifically, the hybrid video encoder takes as input a hybrid image sequence and outputs a visual embedding representing the input vision content. At the same time, the language encoder encodes the sentence into a text embedding that shares a joint embedding space with the visual embedding. Thanks to the large-scale pre-training on the proposed HD-VILA-100M dataset, the multi-modal encoders are able to provide vision-aware text embedding and text-aware vision embedding, which benefits downstream generation tasks. We denote the visual embedding and the text embedding as $\mathbf{v}, \mathbf{t} \in \mathbb{R}^{1024}$ respectively.

Visual/Text Mappers Since the output embedding $\mathbf{v}, \mathbf{t} \in \mathbb{R}^{1024}$ of multi-modal encoders and the latent codes $w^+ \in \mathbb{R}^{18\times512}$ used for generation lie in different feature spaces, we build a visual mapper and a text mapper to bridge the gap between different feature spaces. Specifically, the mapping f is implemented using several layers MLP. It maps the embedding $\mathbf{v}, \mathbf{t}$ to $\mathbf{w}_\mathbf{v}^+, \mathbf{w}_\mathbf{t}^+ \in \mathbb{R}^{18\times512}$,

$$\mathbf{w}_\mathbf{v}^+ = f_v(\mathbf{v}), \quad \mathbf{w}_\mathbf{t}^+ = f_t(\mathbf{t}), \tag{1}$$

where f_v, f_t denote the mapping functions.

Generator (StyleGAN) Since StyleGAN has shown high-fidelity generation quality and impressive disentanglement property, we follow previous works to leverage a StyleGAN for generation [5, 13, 19]. Specifically, we incorporate a well pre-trained and fixed StyleGAN to generate images from the latent codes from mappers $\mathbf{w}_\mathbf{v}^+$ and $\mathbf{w}_\mathbf{t}^+$.

In practice, the latent code $\mathbf{w}_\mathbf{v}^+$ is optimized to reconstruct the high-quality middle frame in the input hybrid image sequence, while the latent code $\mathbf{w}_\mathbf{v}^+$ is optimized to learn the editing directions according to the input sentences. Such a design enables keeping the information from visual inputs, as well as generating novel visual results according to the text inputs. We denote the reconstructed output and the text-guided output as:

$$\mathbf{I}_{rec} = G(\mathbf{w}_\mathbf{v}^+), \tag{2}$$

$$\mathbf{I}_{edit} = G(\mathbf{w}_\mathbf{v}^+ + \mathbf{w}_\mathbf{t}^+), \tag{3}$$

where G denotes the synthesis network of StyleGAN.

D.3.2 Optimization Objectives

To ensure per-pixel reconstruction accuracy, high-quality visual generation, identity preservation, and matching with the descriptions of the generated results, we carefully select a pixel-wise $\mathcal{L}_2$ loss, a LPIPS loss [24], an identity loss [14], and a text-visual matching loss as our optimization objectives following common practices [13, 14, 18, 19]. Specifically, the pixel-wise ℓ_2 loss is denoted as:

$$\ell_2(\mathbf{I}, \hat{\mathbf{I}}) = ||\mathbf{I} - \hat{\mathbf{I}}||_2, \tag{4}$$

where $\mathbf{I}$ denote the high-quality middle frame. LPIPS is a deep metric that is able to reflect image quality similar to human perceptual [24], and the LPIPS loss is denoted as:

$$\ell_{lpips}(\mathbf{I}, \hat{\mathbf{I}}) = ||\mathcal{F}(\mathbf{I}) - \mathcal{F}(\hat{\mathbf{I}})||_2, \tag{5}$$

where $\mathcal{F}$ denotes the perceptual feature extractor. We follow Richardson et al. to incorporate an identity recognition loss to measure the cosine similarity between the output image and its target [14],

$$\ell_{id}(\mathbf{I}, \hat{\mathbf{I}}) = 1 - \left\langle \mathcal{R}(\mathbf{I}), \mathcal{R}(\hat{\mathbf{I}}) \right\rangle, \tag{6}$$

where $\mathcal{R}$ is a pre-trained network for face feature extractor, and $\langle \cdot, \cdot \rangle$ denotes cosine similarity calculation. To ensure the matching between the text-guided output and the input text, we follow StyleCLIP [13] to include a matching loss for optimization. In particular, the matching loss aims at minimizing the feature distance between the output image and the text,

$$\ell_{clip}(\mathbf{T}, \hat{\mathbf{I}}) = 1 - \left\langle \mathcal{C}(\mathbf{T}), \mathcal{C}(\hat{\mathbf{I}}) \right\rangle / \gamma, \tag{7}$$

where $\mathcal{C}$ is a pre-trained image-text feature extractor, $\mathbf{T}$ denotes the text input, and γ is a constant value that normalize the similarity value to the range of [0,1]. In practice, we set the value of γ as 100. The overall optimization objectives are concluded as:

$$\begin{aligned}\ell = \lambda_1 \cdot \ell_2(\mathbf{I}, \mathbf{I}_{rec}) + \lambda_2 \cdot \ell_{lpips}(\mathbf{I}, \mathbf{I}_{rec}) + \lambda_3 \cdot \ell_{id}(\mathbf{I}, \mathbf{I}_{rec}) \\ + \lambda_4 \cdot \ell_2(\mathbf{I}, \mathbf{I}_{edit}) + \lambda_5 \cdot \ell_{lpips}(\mathbf{I}, \mathbf{I}_{edit}) + \lambda_6 \cdot \ell_{id}(\mathbf{I}, \mathbf{I}_{edit}) \\ + \lambda_7 \cdot \ell_{clip}(\mathbf{T}, \mathbf{I}_{edit}).\end{aligned} \tag{8}$$

Implementation Details We empirically set the loss weights for different generation tasks. For text-guided editing, we set $\lambda_1 = 1.0, \lambda_2 = 0.8, \lambda_3 = 0.1, \lambda_4 = 0.1, \lambda_5 = 0.1, \lambda_6 = 0.1, \lambda_7 = 1.0$. For text-guided super-resolution, we set $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0.1, \lambda_5 = 0.8, \lambda_6 = 1.0, \lambda_7 = 0.1$. We use a fixed learning rate $1e-5$ for the training of the multi-modal encoders, and a fixed learning rate $1e-3$ for the visual/text mappers. We use Adam optimizer with $(\beta_1, \beta_2) = (0.9, 0.99)$ for training [6]. We train the models for 200K iterations in total on 4 NVIDIA Tesla V100 GPUs.

D.3.3 Face-Description-Video Dateset (FDVD)

To demonstrate the effectiveness of our text-guided generation framework on videos, we collected a dataset of video-description pairs of the human faces, named **Face-Description-Video Dateset (FDVD)**. FDVD consists of 613 video-description pairs, resulting in 74,803 frames of human faces and 6,130 sentences in total. Specifically, each video-description pair consists of one high-resolution video (1024×1024 spatial size) and ten different descriptive sentences. We introduce the collection process as below.

To generate high-quality videos of human faces, we collected videos from Ryerson audio-visual dataset [10]. For the pre-processing, we first use the facial landmark locations to select an appropriate crop region for the talking head, then we perform a high-quality up-sampling to obtain the final videos at 1024×1024 resolution following [4]. To generate diverse descriptions for each video, we adopt a strategy of *prediction-and-generation*. First, we use a facial attribute predictor [9] to obtain a list of attributes for the videos. Then we follow previous best practices to use PCFG rule-based algorithm to generate descriptions from the given attributes [17, 19]. Each description contains different subsets of the attributes to increase the diversity of descriptions. We will release the dataset for research purposes.

D.3.4 Experiments

Text-Guided Editing To demonstrate the effectiveness of our text-guided generation framework, we show the qualitative comparison of text-guided editing results in Figure 5. Specifically, we compare our full model with the one without pre-training, StyleCLIP [13] and TediGAN [19]. StyleCLIP and TediGAN are two state-of-the-art text-guided editing approaches. Both of them combine the strong generative powers of StyleGAN with text input for editing. Specifically, StyleCLIP maps a text prompt into an input-agnostic direction in StyleGAN's style space [13], and TediGAN proposes to map the text into StyleGAN's style space directly. We use the released code provided by the authors on their official homepage to obtain the results in Figure 5.

The results in Figure 5 show that our pre-trained model can benefit the downstream text-guided editing task and achieve state-of-the-art performance. Take the first case as an example, our model without pre-training tends to make the lips bigger when it wears the lipstick, and StyleCLIP and TediGAN fail to attend to the keyword "eyeglasses" in the natural but relatively complex descriptions. Thanks to the power of our pre-trained model, our generation framework is able to attend to multiple attributes and edit the images accurately.

We also provide **a video demo** generated by our full model in this supplementary material (**video.mp4**). The video demo consists of 10 video cases. In each case, we show the input on the left, our result on the right, and the input description on top. We take as input a video clip and a target description as input and generate the videos frame-by-frame. The video demo shows that our model shows promising text-guided video editing performance.

Figure 5. **Qualitative comparison of text-guided editing results**. We show from left to right the inputs, results of our full model, results of our model without pre-training, results of StyleCLIP [13] and TediGAN [19]. The comparison shows that our pre-trained model can benefit the downstream text-guided editing task and achieve state-of-the-art performance. Due to the vision-aware text embedding learned from pre-training, our full model is able to attend to "She", "eyeglasses" and "rep lipstick" in the first case and accurately edit the images accordingly.

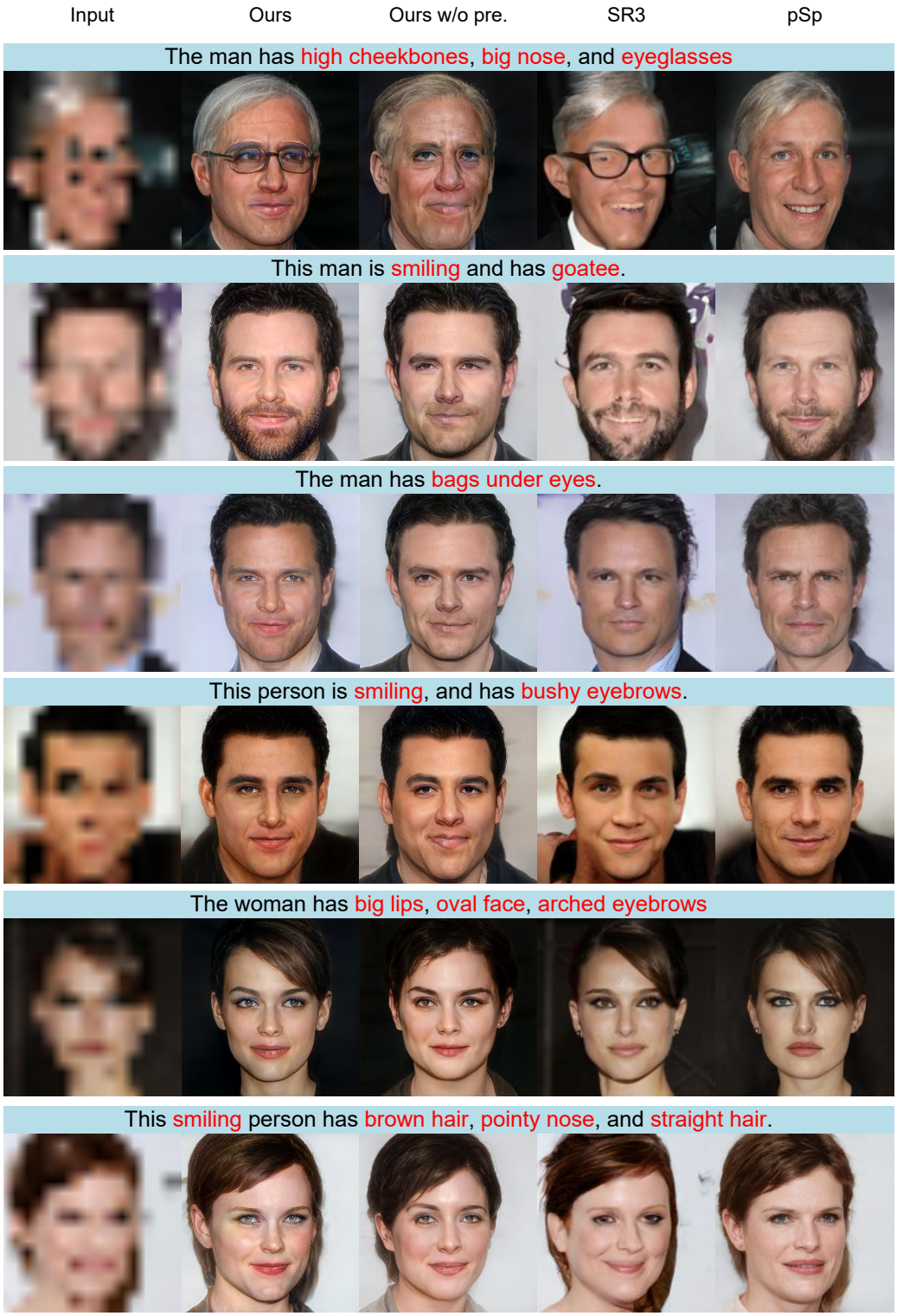


Figure 6. **Qualitative comparison of more super-resolution results**. We show from left to right the inputs, results of our full model, results of our model without pre-training, results of SR3 [16] and pSp [14]. Our pre-trained model could generate realistic results with more textual attributes (*e.g.*, eyeglasses in the first example, big lips and arched eyebrows in the 5-th example) due to the power of our pre-trained model.

Text-Guided Super-Resolution The results of the super-resolution task are presented in Figure 6. We take relative low resolution images (16×16) as input(1-st column) and generate high-resolution results (2-nd column). We train our framework from scratch and present the results in 3-rd column, which fail to capture some textual information. Two other strong baselines we adopt are SR3 [16] and pSp [14]. Their results are presented in the 4-th and 5-th colomn respectively. Our pre-trained model could generate realistic results with more textual attributes (e.g., eyeglasses in the first example, big lips and arched eyebrows in the 5-th example). Super-resolution is an ill-posed problem, which means a low-resolution image may be downsampled from different high-resolution images. The details can't be well constructed with our text. How to keep the consistency between consensus frames and save the details (*e.g.*, hair) are still worth exploration. Besides, the pre-trained and fixed StyleGAN [5] are trained on a dataset with specific distribution and may introduce bias. With our general and diverse data, we hope we could alleviate the problem in the future.

D.4. Ablation Study on Data Domain

Methods	Steps	R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓
HowTo100M [37]	-	8.2	24.5	35.3	24.0
Ours (HowTo100M)	145K	15.7	38.3	51.3	10.0
Ours (HD-VILA-100M)	145K	6.6	19.5	27.6	37.0
Ours (HD-VILA-100M)	504K	9.1	25.5	37.3	20.0

Table 3. Comparison of pre-training datasets on YouCook2 retrieval

We conduct text-to-video retrieval task on YouCook2 [25] to check whether pre-training with in-domain dataset could benefit downstream tasks. We can see that although our model pre-trained on HD-VILA-100M outperforms HowTo100M model in their paper, our model pre-trained on HowTo100M performs best in limited epochs. This shows pre-training on in-domain dataset could benefit VL tasks very much.

E. Datasheet for HD-VILA-100M

In this section, we provide a DataSheet [2] for HD-VILA-100M.

E.1. Motivation

- **For what purpose was the dataset created?** We provide this dataset in order to explore multi-modality representation learning with large scale of video-language data available in the Internet. Previous datasets are limited in scale and diversity. A large-scale video-language dataset is crucial for the research community.
- **Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?** This dataset was created by Microsoft Research Asia.

E.2. Composition

- **What do the instances that comprise the dataset represent?** The instances of this dataset are video and each video is paired with ASR transcripts aligned over time.
- **How many instances are there in total?** We include 3.3 million videos. Altogether, we extracted 103 million video clips with ASR transcripts from this data.
- **Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?** It is a sample. We only keep videos with quality higher or equal to 720p from the YouTube website. The dataset covers 15 popular categories with a wide range of topics from YouTube to make it more representative.
- **What data does each instance consist of?** The instance consist of a short video clip with an average duration of 13.4 seconds and an ASR transcript with 32.5 words in average.
- **Is there a label or target associated with each instance?** We use ASR transcripts as the labels of video clips in this dataset.
- **Is any information missing from individual instances?** No.
- **Are relationships between individual instances made explicit?** Not applicable. The relationship between videos is not the focus in our study, though it could be possible for future work.
- **Are there recommended data splits?** No. We build this dataset only for pre-training so we have not created validation set this time.
- **Are there any errors, sources of noise, or redundancies in the dataset?** Yes. The ASR transcripts are often noisy with mistakes. Although we use some methods to clean the data, there are still errors we cannot fix.
- **Is the dataset self-contained, or does it link to or otherwise rely on external resources?** The dataset is self-contained. However, we plan to only release the URLs of videos and the code for preparing data. This can protect user privacy in case some videos will be deleted by YouTube users.

- **Does the dataset contain data that might be considered confidential?** No. We only contain videos that are public to everyone on YouTube.
- **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** Yes, some videos in the YouTube are. We try our best to decrease the number of offensive videos by avoiding offensive topics.
- **Does the dataset identify any subpopulations (e.g., by age, gender)?** Not explicitly (e.g., through labels).
- **Is it possible to identify individuals, either directly or indirectly from the dataset?** Yes, our data includes celebrities, or other YouTube-famous people. All of the videos that we use are of publicly available data, following the Terms of Service that users agreed to when uploading to YouTube.
- **Does the dataset contain data that might be considered sensitive in any way?** Yes, some of YouTube videos might be. We try to avoid this by removing sensitive topics.

E.3. Collection Process

- **How was the data associated with each instance acquired?** The dataset is directly observable from YouTube.
- **What mechanisms or procedures were used to collect the data?** We collect the dataset using YouTube API and youtube-dl tool.
- **If the dataset is a sample from a larger set, what was the sampling strategy?**

  We use a probabilistic sampling strategy to cover more categories and make the dataset more balanced. More details can be found in **Section 3 Dataset** in the main paper.
- **Who was involved in the data collection process and how were they compensated?** Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu are mainly responsible for data collection. The other authors are also involved in discussing the data collection process.
- **Over what timeframe was the data collected?** This dataset was collected from September 2021 to October 2021, although the YouTube are often much older (dating back to when the platform was first created).
- **Were any ethical review processes conducted ?** There is no official processes conducted, since we create this dataset for research without human subjects.

E.4. Preprocessing/cleaning/labeling

- **Was any preprocessing/cleaning/labeling of the data done?** Yes. We process the ASR transcriptions and cut the videos into clips. More details can be found in **Section 3 Dataset** of the main paper.
- **Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?** Yes, but we do not plan to release the "raw" data due to copyright and privacy concerns.
- **Is the software that was used to preprocess/clean/label the data available?** Yes. We use an off-the-shelf tool to process ASR transcriptions, it can be found at here [1]. The other code used for processing the data will also be released.

E.5. Uses

- **Has the dataset been used for any tasks already?** If so, please provide a description. At the time of data release, only our paper has used it.
- **Is there a repository that links to any or all papers or systems that use the dataset?** No.
- **What (other) tasks could the dataset be used for?** This dataset can be used for general video-language pre-training and the pre-trained model can be transferred to a wide range of downstream tasks, e.g., video-text retrieval, video QA, video captioning.
- **Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?** Since we only release the URLs of the videos, there might be some videos missing in the future due to deleting by YouTube users or YouTube website.
- **Are there tasks for which the dataset should not be used?** This dataset is created for research instead of commercial usage. Tasks that are sensitive or offensive should not use this dataset.

E.6. Distribution

- **Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?** We will release the dataset to public.
- **How will the dataset will be distributed?** The dataset will be distributed in GitHub [2]. We will only release

[1] https://github.com/ottokart/punctuator2

[2] https://github.com/microsoft/XPretrain/tree/main/hd-vila-100m

the URLs of the videos and some meta-data (e.g., time span of video clips).

- **When will the dataset be distributed?** The dataset will be released by March 28, 2022.
- **Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?** The dataset is under the Open Use of Data Agreement (O-UDA) [3].
- **Have any third parties imposed IP-based or other restrictions on the data associated with the instances?** No.
- **Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?** No.

E.7. Maintenance

- **Who will be supporting/hosting/maintaining the dataset?** All the corresponding authors of this work.
- **How can the owner/curator/manager of the dataset be contacted?** By emailing the contact persons in the release page.
- **Is there an erratum?** No.
- **Will the dataset be updated?** We do not plan to update it at this time.
- **Will older versions of the dataset continue to be supported/hosted/maintained?** This is the first version of this dataset.
- **If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?** No at this time.

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, pages 5803–5812, 2017. 4

[2] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021. 9

[3] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, pages 2758–2766, 2017. 2

[4] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 6

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 5, 9

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[7] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, pages 706–715, 2017. 4

[8] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *BMVC*, 2019. 4

[9] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 6

[10] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, page e0196391, 2018. 6

[11] 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*, pages 9879–9889, 2020. 4

[12] 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*, pages 2630–2640, 2019. 1, 2

[13] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, pages 2085–2094, 2021. 5, 6, 7

[14] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *CVPR*, pages 2287–2296, 2021. 5, 6, 8, 9

[15] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Joseph Pal, H. Larochelle, Aaron C. Courville, and Bernt Schiele. Movie description. *IJCV*, pages 94–120, 2016. 4

[16] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*, 2021. 8, 9

[17] David Stap, Maurits Bleeker, Sarah Ibrahimi, and Maartje ter Hoeve. Conditional image generation and manipulation for user-specified content. In *CVPR Workshop*, 2020. 6

[18] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *TOG*, 40(4):1–14, 2021. 5

[19] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *CVPR*, pages 2256–2265, 2021. 5, 6, 7

[3] https://github.com/microsoft/Open-Use-of-Data-Agreement

[20] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM MM*, page 1645–1653, 2017. 2

[21] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296, 2016. 2, 4

[22] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, pages 471–487, 2018. 2, 4

[23] Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *ECCV*, pages 374–390, 2018. 4

[24] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 5, 6

[25] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 9