A. Additional Results

Fair Comparison to VIOLET [8]. VIOLET proposes to augment VTM+MLM with masked visual token modeling for VidL pre-training, while only showing marginal improvements on downstream performance. In contrast, we conduct comprehensive investigations across different MVM targets and masking strategies to demonstrate that effective MVM training can largely improve downstream performance. Note that our study already encompasses the design of MVM in [8], that is MVM with VQ target and BM+AM as the masking strategy. To make a fair comparison between [8] and our best setting (MVM with SIF target and BM+AM as the masking strategy), we reproduce [8] under the same setting and report downstream performance in Table 1. Results show that our setting obtains a significant improvement, with +2.3% on TGIF-Frame and +13.3% on AveR for DiDeMo-Retrieval, respectively. These results suggest the importance of an appropriate MVM setting, which is the core belief in our study.

Method	MVM Target	TGIF-Frame Acc.	DiDeMo-Retrieval R1	R5	R10	AveR
VIOLET [8]	VQ	70.5	32.9	63.0	74.5	56.8
VIOLETv2	SIF	**72.8**	**47.9**	**76.5**	**84.1**	**69.5**

Table 1. Fair comparison to VIOLET [8]. Both models are pre-trained on WebVid [2]+CC [27].

MVM *vs.* Temporal self-supervised objectives. In addition to the reconstructive MVM task, other self-supervised video modeling tasks can be explored, for example, Frame Order Modeling (FOM) [12, 34], which reconstructs the temporal orders of shuffled frame inputs. In Table 2, we compare MVM (SIF) with FOM in [34], when pre-trained on WebVid [2]. MVM (SIF) still leads to better performance, with a gain of +0.7% on TGIF-Frame and +4.8% on AveR for DiDeMo-Retrieval, respectively.

VTM+MLM+	TGIF-Frame Acc.	DiDeMo-Retrieval R1	R5	R10	AveR
MVM (SIF)	**68.8**	**35.1**	**63.3**	**73.1**	**57.2**
FOM	68.1	27.6	59.0	70.7	52.4

Table 2. MVM vs. FOM. Both models are pre-trained on WebVid [2].

Initialization and Learning of Video Backbone. We investigate the effect of different initialized video backbones with or without MVM on VL inputs in Table 3. At first, although the used video transformer (VT) is randomly initialized, the MVM training still enhances the visual representation and benefits the downstream video-language (VidL) tasks. Furthermore, MVM can also boost better initialized VT from VidSwin-B and lead to a comprehensive increase. Specifically, the improvement gap is more significant than random initialization, where we can learn better from MVM and enlarge its effectiveness during pre-training. We additionally compare two self-supervised initializations with MVM on video-only inputs, one with TVF as the MVM target and the other with SIF. Though VidL pre-training with MVM from supervised VidSwin-B initialization leads to the best downstream performance, we observe consistent performance improvement from MVM on VL inputs regardless of the initialization setting.

Weight Init.	MVM on VL	TGIF-Frame Acc.	DiDeMo-Retrieval R1	R5	R10	AveR
Random	✗	55.9	5.6	19.9	29.8	18.5
	✓	56.5	7.4	22.9	33.8	21.4
V-only MVM (TVF)	✗	59.9	15.3	38.4	54.7	36.1
	✓	60.2	17.4	43.2	56.0	38.9
V-only MVM (SIF)	✗	61.0	16.9	42.4	54.9	38.1
	✓	61.5	18.6	44.0	58.1	40.2
VidSwin-B	✗	68.1	28.7	57.0	69.7	51.8
	✓	**68.8**	**35.1**	**63.3**	**73.1**	**57.2**

Table 3. Impact of **weight initialization and learning of video backbone**. All variants are pre-trained on video-text from WebVid [2] for 5 epochs. The MVM target is spatial-focused image features (SIF) from Swin-B [16]), if not specified otherwise. For V-only MVM (TVF/SVF), we first self-supervisedly pre-train the video backbone with MVM on video-only inputs from WebVid for 5 epochs. The final pre-training setting is highlighted in gray.

Type of MVM Loss. We compare the type of loss function for the MVM training by using least absolute deviations (l_1) or least square errors (l_2) in Table 4. It is well known that the l_1 loss can be resistant to outlier data. We show that MVM through l_1 is also more robust and leads to better performance on both video question answering and text-to-video retrieval than the l_2 loss.

MVM Loss	TGIF-Frame Acc.	DiDeMo-Retrieval R1	R5	R10	AveR
l_1	**68.8**	**35.4**	**62.4**	**74.9**	**57.6**
l_2	**68.8**	33.0	60.1	71.9	55.0

Table 4. Impact of **MVM loss type**. All variants are pre-trained on WebVid [2] with VTM+MLM+MVM (SIF) for 5 epochs, using RM as the masking strategy with ratio of 15%. The final pre-training setting is highlighted in gray.

MVM Prediction Head. We investigate the prediction head for MVM in Table 5. As a result, a single linear layer is not enough to model the complicated distilling MVM features. (*e.g.,* 31.3 *vs.* 35.4 R1 on DiDeMo-Retrieval) Therefore, we follow VTM and MLM to use 2-layer MLP as the prediction head for MVM.

MVM Head	TGIF-Frame Acc.	DiDeMo-Retrieval R1	R5	R10	AveR
1 Linear Layer	**68.8**	31.3	60.1	72.8	54.7
2-layer MLP	**68.8**	**35.4**	**62.4**	**74.9**	**57.6**

Table 5. Impact of **MVM prediction head**. All variants are pre-trained on WebVid [2] with VTM+MLM+MVM (SIF) for 5 epochs, using RM as the masking strategy with ratio of 15%. The final pre-training setting is highlighted in gray.

TVF Target Extractors vs. Downstream Performance. We compare distilling video features from VidSwin-B vs. VidSwin-L (the default setting in the main text) in Table 6.

Here, for experiments with VidSwin-B, the same VidSwin-B weight is used to initialize the video backbone and to extract the MVM target. Hence, the MVM objective can be easily minimized by simply ignoring the text inputs, which conflict with the other objectives. This variant is in principle similar to masked frame modeling in HERO [12], the key difference lies in whether the video backbone is refined during pre-training. In addition, we investigate whether the sparse sampling of video frames when extracting TVF target is the key reason behind the lower performance of TVF, compared to SVF. Hence, we compare the default sparse sampling of 5 frames, against a dense-version of TVF target (feeding 16 frames into VidSwin-L). While the dense input is slightly beneficial, SIF still performs better, with absolute advantages of +0.4% on TGIF-Frame and +1.8% on AveR for DiDeMo-Retrieval.

MVM Target	TGIF-Frame	DiDeMo-Retrieval			
	Acc.	R1	R5	R10	AveR
TVF (VidSwin-L [17])	68.0	32.8	60.5	73.0	55.4
TVF (VidSwin-B)	67.5	25.8	55.0	68.0	49.6
TVF-dense (VidSwin-L)	**68.4**	**34.3**	**60.8**	**72.4**	**55.8**

Table 6. Temporal-aware video feature (TVF) target models *vs.* downstream performance. All variants are pre-trained on WebVid [2] with VTM+MLM+MVM (TVF) for 5 epochs, using RM as the masking strategy with ratio of 15%.

Additional Exploration in Combining MVM Targets. We explore the additional combination of distilling MVM targets in Table 7. MVM with SIF has an obvious advantage over TVF only on both video question answering and text-to-video retrieval. While, considering SIF+TVF seems not to bring a robust improvement, especially decreasing text-to-video retrieval. The previous study [3] shows that current VidL benchmarks primarily focus on spatial understanding of the key frame from videos. Furthermore, combining TVF with SIF results in excessive training overhead. Accordingly, we choose SIF as our final pre-training setting.

MVM Targets	TGIF-Frame	DiDeMo-Retrieval			
	Acc.	R1	R5	R10	AveR
SIF	68.8	**35.4**	62.4	**74.9**	**57.6**
TVF	68.0	32.8	60.5	73.0	55.4
SIF + TVF	**69.2**	33.8	**63.0**	74.4	57.1

Table 7. Combining target features for MVM. All variants are pre-trained on WebVid [2] for 5 epochs. The final pre-training setting is highlighted in gray.

Additional Exploration of SIF Target. We explore a more advanced SIF target, DeiT [30], in Table 8. These results show that Swin-B still has an advantage (a noticeable higher 34.9 R1 on retrieval), consistent with our previous observations in the main text. That is, SIF should share similar inductive biases to the video encoder (i.e., Swin-T/B/L).

Investigation of Training Recipe with CLIP Target. We presented initial results for varying training settings using CLIP/Swin-B targets and compare them with the default setting in Table 1. Swin-B had a significant advantage over CLIP. As we adjust the training recipe with CLIP target in Table 9, a better training recipe reduces the performance gap between the SIF target and the CLIP target. The results in turn suggest the importance of an effective MVM strategy (e.g., masking ratio). Though impossible to iterate over all settings, Swin-B remains competitive under the same training recipe, especially with limited training data (IN-22K vs. 400M). Note that we use CLIP image features as the MVM target, while other related works [15, 20] use them as model inputs. One potential enhancement is to leverage the multimodal information from both image and text encoders in CLIP (similar to the use of both in [20]), which is an interesting direction to explore in future studies. However, our setup is still valid, as we aim to train a fusion-encoder architecture rather than a dual-encoder architecture as CLIP.

Image Feat. Model	Train Data	IN-1K ACC@1	TGIF-Frame Acc.	DiDeMo-Retrieval R1	R5	R10
ResNet-50	IN-1K	76.1	67.3	29.1	58.1	69.3
DeiT [30]	IN-1K	83.4	**68.4**	31.4	59.4	72.2
Swin-B	IN-1K	**83.5**	68.3	**34.9**	**63.4**	**73.9**

Table 8. Comparing Swin-B vs. another SIF model (DEiT) All variants are pre-trained on WebVid with VTM+MLM+MVM (SIF) for 5 epochs, using RM with 15% as the masking strategy.

MVM target	Settings	TGIF-Frame Acc.	DiDeMo-Retrieval R1	R5	R10
CLIP	Default	67.7	29.8	57.8	68.5
Swin-B	Default	68.8	35.1	63.3	73.1
CLIP	lr × 2	70.5	32.9	61.6	73.5
CLIP	masking ratio = 0.3	68.0	31.8	59.6	71.3
CLIP	loss type = l_2	68.3	30.1	59.1	71.0
CLIP	linear MVM head	68.2	30.5	58.3	69.2
Swin-B	lr × 2	**70.6**	33.3	63.7	**75.2**
Swin-B	masking ratio = 0.3	68.8	**36.2**	**64.0**	74.5

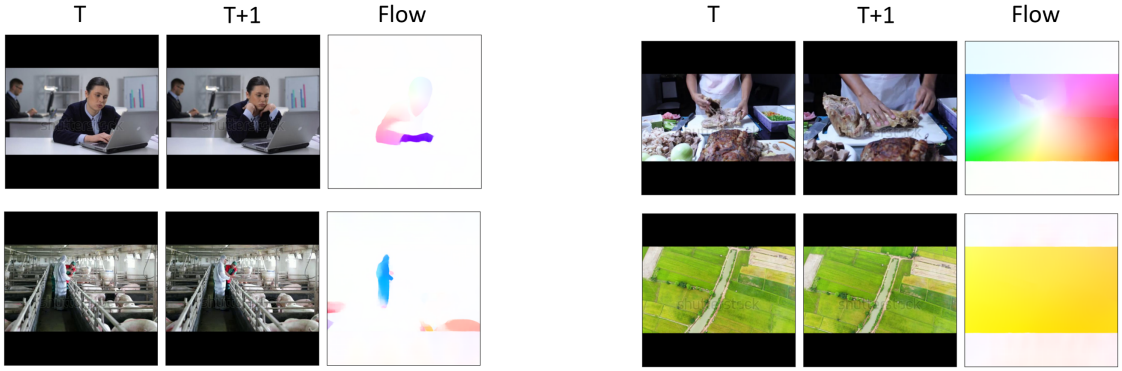
Table 9. Investigation of Training Recipe with CLIP Target. All variants are pre-trained on WebVid with VTM+MLM+MVM for 5 epochs. The default setting follows Table 1 in the main text, that is RM with 15% masking ratio, l_1 loss and 2-layer MLP head for MVM prediction.

Extended Results for Table 1. The additional results on MSVD-QA and MSRVTT-Retrieval in Table 10 show that *SIF* is still the most effective, consistent with Table 1. However, the effects of different MVM targets seem to be better on MSVD-QA and MSRVTT-Retrieval (on average 15s long) than those on TGIF-Frame (on average 3s long) and DiDeMo-Retrieval (on average 30s long). We hypothesize that different downstream video lengths may contribute to different performance gains/losses when evaluating the effectiveness of an MVM target, which we leave as future directions for investigation.

Qualitative Results. Figure 1 shows good and bad examples of optical flow predictions made by RAFT-L [28] with sparsely sampled frames. As shown in 1b, the top example shows zoom-in shots, and the bottom shows moving shots. All content in the current frame is moving, which is the

Pre-training Tasks	MVM Target	MSVD-QA	MSRVTT-Retrieval			
		Acc.	R1	R5	R10	AveR
VTM+MLM	None	49.2	26.0	56.6	69.4	50.7
+MVM	RGB Pixel Values	51.0 (+1.8)	27.4 (+1.4)	58.0 (+1.4)	69.8 (+0.4)	51.7 (+1.0)
	Histogram of Oriented Gradients [6]	50.1 (+0.9)	27.4 (+1.4)	57.7 (+1.1)	70.2 (+0.8)	51.8 (+1.1)
	Depth Maps (DPT-L [24])	50.3 (+1.1)	28.0 (+2.0)	57.4 (+0.8)	70.6 (+0.8)	52.0 (+1.3)
	Optical Flow (RAFT-L [28])	49.7 (+0.5)	25.8 (-0.2)	55.8 (-0.8)	69.4	50.3 (-0.4)
	Spatial-focused Image Features (Swin-B [16])	**51.1 (+1.9)**	29.4 (+3.4)	**59.9 (+3.6)**	**73.1 (+3.7)**	**54.1 (+3.4)**
	Temporal-aware Video Features (VidSwin-L [17])	49.8 (+0.6)	29.9 (+3.9)	58.1 (+1.5)	70.2 (+0.8)	52.7 (+2.0)
	Discrete Visual Tokens (DALL-E [23])	50.7 (+1.5)	27.3 (+1.3)	58.3 (+1.7)	70.0 (+0.6)	51.9 (+1.2)
	Multimodal Features (CLIP-ViT-B [22])	50.2 (+1.0)	**30.0 (+4.0)**	58.8 (+2.2)	71.1 (+1.7)	53.3 (+2.6)

Table 10. Comparing target features for MVM applied to video-text data. All variants are pre-trained on WebVid [2] for 5 epochs. Masking is performed randomly (RM) with a ratio of 15%. The final pre-training setting is highlighted in gray.



(a) **(b)**

Figure 1. Visualization of optical flow (Flow) predictions by RAFT-L [28] with sparsely sampled frames. We show examples of good cases in (a) and bad cases in (b).

main reason behind the failure in optical flow estimation.

We also show the visualizations of zero-shot text-to-video retrieval on MSRVTT (Figure 2), DiDeMo (Figure 3), and LSMDC (Figure 4) to demonstrate that MVM can help video understanding from different domains, such as gaming, animation, human activity, or movie scene.

B. Additional Pre-training Details

Vidoe-Text Matching (VTM). VTM enhances the cross-modal fusion via modeling the alignments between visual and textual inputs. At each training step, we randomly replace the corresponding text $\mathcal{X}_{\text{pos}}$ for a given video $\mathcal{V}$ with the text description $\mathcal{X}_{\text{neg}}$ from a different video in the same batch. Both the positive pair $(\mathcal{V}, \mathcal{X}_{\text{pos}})$ and negative pair $(\mathcal{V}, \mathcal{X}_{\text{neg}})$ are modeled by Cross-modal Transformer (CT), and VTM is to tell them apart from the global VidL representation h^{c} of the [CLS] token. In particular, h^{c} will be processed by a fully-connected layer (FC^{VTM}) to learn contrastively through classification:

$$b^{\text{pos}} = \text{FC}^{\text{VTM}}(h^{\text{c}}_{\text{pos}}), b^{\text{neg}} = \text{FC}^{\text{VTM}}(h^{\text{c}}_{\text{neg}}),$$
$$\mathcal{L}_{\text{VTM}} = -\frac{1}{B}\sum_{i}^{B} \log \frac{b^{\text{pos}}_i}{b^{\text{pos}}_i + \sum b^{\text{neg}}_i}, \quad (1)$$

where $h^{\text{c}}_{\text{pos}}$ or $h^{\text{c}}_{\text{neg}}$ is h^{c} of positive or negative pairs.

Masked Language Modeling (MLM). In MLM, we randomly mask out some word tokens with a probability of 15%.[1] The goal is to recover these masked word tokens x from the joint VidL features h modeled by CT. Specifically, the corresponding h^{x} for these masked tokens are fed in a fully-connected layer (FC^{MLM}) and projected to the discrete token space for classification:

$$x'_i = \text{FC}_{\text{MLM}}(h^{\text{x}}_i),$$
$$\mathcal{L}_{\text{MLM}} = -\mathbb{E}\,[\frac{1}{|\mathcal{M}^{\text{MLM}}|}\sum_{i\in\mathcal{M}^{\text{MLM}}} \log P(x_i \mid x'_i)], \quad (2)$$

where $\mathcal{M}^{\text{MLM}}$ denotes the index set of masked word tokens.

Implementation Details. Our VIOLETv2 is implemented based on PyTorch [21]. As discussed in the main text and supported by the additional experimental results above, our final pre-training setting is (i) VTM+MLM+MVM (with MVM target as spatial-focused image features from Swin-B [16], applied on video-text inputs only) as the pre-training tasks; (ii) 2-layer MLP as the MVM prediction head and l_1 regression as the MVM loss; and (iii) blockwise masking + random masking with masking ratio of 30% as the masking strategy. We adopt AdamW [18] as the optimizer with a warmup learning rate schedule of 5e-5 peak learning rate, betas of (0.9, 0.98), and weight decay of 1e-3 for all pre-training experiments. We pre-train our model on 32

[1] Following BERT [7], We replace 80% of masked word tokens as the [MASK] token, 10% as a random token, and 10% as its original token.

NVIDIA V100 GPUs with a batch size of 28 per GPU. Pre-training with 10 epochs on WebVid2.5M [2] + CC3M [27] takes about 27 hours to finish. We present the training settings for all finetuning experiments in the next section.

C. Experimental Setup of Downstream Tasks

We test our pre-trained models on 3 popular VidL tasks across 13 downstream datasets, including text-to-video retrieval, video question answering, and video captioning. For text-to-video retrieval, we report downstream performance on MSRVTT [32], DiDeMo [9], and LSMDC [26] and use Recall at K (R@K, K=1,5,10) as the evaluation metric. For video question answering, we consider datasets in both multiple-choice and open-ended settings, including TGIF-Action, TGIF-Transition, TGIF-Frame [10], MSRVTT-MC [33], MSRVTT-QA, MSVD-QA [31], LSMDC-MC and LSMDC-FiB [29]. We evaluate our models using accuracy. For video captioning, we report CIDER scores on MSRVTT and MSVD.

We follow the standard training/validation/testing splits of the original datasets. If not otherwise stated, we sparsely sample $T = 5$ video frames and adopt video frame size 224 with patch size $H = W = 32$. Similar to pre-training, we use AdamW [18] to fine-tune our model for each downstream task with a warmup learning rate schedule of 2e-5 peak learning rate, betas of (0.9, 0.98), and weight decay of 1e-3. All finetuning experiments are conducted on Microsoft Azure [1] adopting mixed-precision training with DeepSpeed [25].[2] All video data are pre-processed by evenly extracting 32 frames to avoid expensive decoding on-the-fly. During training, we randomly sample T frames from 32 frames, resize the shorter side of all frames to 224, and random crop (224x224) at the same location for all the frames in a given video. During inference, we evenly sample T frames from 32 frames and center crop (224x224) for all the sampled video frames.

C.1. Text-To-Video Retrieval

For text-to-video retrieval, similar to visual-text matching (VTM) during pre-training, we treat corresponding video-text pairs in the same batch as positives and all other pairwise combinations as negatives. We adopt a fully-connected (FC) layer (FC^{T2V}) over the VidL representation h^{c} of the `[CLS]` token to learn through classification:

$$b^{\text{pos}} = \text{FC}^{\text{T2V}}(h^{\text{c}}_{\text{pos}}), b^{\text{neg}} = \text{FC}^{\text{T2V}}(h^{\text{c}}_{\text{neg}}),$$
$$\mathcal{L}_{\text{T2V}} = -\frac{1}{B}\sum_{i}^{B} \log \frac{b^{\text{pos}}_i}{b^{\text{pos}}_i + \sum b^{\text{neg}}_i}, \quad (3)$$

where $h^{\text{c}}_{\text{pos}}$ or $h^{\text{c}}_{\text{neg}}$ is h^{c} of positive or negative pairs. In particular, we use pre-trained FC^{VTM} for zero-shot text-to-video retrieval and to initialize FC^{T2V} for further fine-tuning on each downstream text-to-video retrieval task.

[2]We conduct retrieval finetuning on 8 80GB A100 GPUs to enable larger batch size, while all other finetuning experiments are conducted on 8 32GB V100 GPUs.

VideoQA	Task	#Option
Multiple-Choice	TGIF-Action [10]	5
	TGIF-Transition [10]	5
	MSRVTT-MC [33]	5
	LSMDC-MC [29]	5
Open-Ended	TGIF-Frame [10]	-
	MSRVTT-QA [31]	-
	MSVD-QA [4]	-
	LSMDC-FiB [29]	-

Table 11. Summary of **video question answering** tasks. For open-ended Video QA, we do not limit the answer vocabulary to a fixed candidate set.

MSRVTT [32] contains 10K YouTube videos with 200K human annotations. For fair comparison [2, 11], we train on 9K training+validation splits and evaluate on the 1K-A testing split. We adopt batch size 20 per GPU and train for 10 epochs.

DiDeMo [9] consists of 10K videos annotated with 40K sentences from Flickr. Following [2, 11], we concatenate all sentences from the same video into a paragraph and perform paragraph-to-video retrieval for DiDeMo. We adopt batch size 16 per GPU and train for 10 epochs.

LSMDC [26] contains 118K video clips from 202 movies. Each clip has a caption from movie scripts or descriptive video services. Following [2, 19], we evaluate on 1K testing clips that disjoint from the training+validation splits. We adopt batch size 20 per GPU and train for 5 epochs.

C.2. Video Question Answering

We test our model on video question answering (QA) tasks in both multiple-choice and open-ended settings as Table 11. We follow LAVENDER [13] to formulate Video QA as Masked Language Modeling due to its superior performance. For multiple-choice QA tasks, we concatenate question with all answer options and add a `[MASK]` to form the input text (`Q+A0+A1+A2+A3+A4+[MASK]`). We treat the same Masked Language Modeling (MLM) layer as used in pre-training upon h^{x} to predict the word token corresponding to the answer index (*e.g.*, `0,1,2,3,4`). Similarly, for open-ended QA tasks, we apply MLM over the input (`Q+[MASK]`). Cross-entropy loss is used to supervise the downstream finetuning over the whole word vocabulary.

TGIF-Action, TGIF-Transition, and TGIF-Frame [10] require spatial-temporal reasoning to answer questions regarding GIF videos in TGIF-QA Specifically, we aim to test our model along three dimensions: (i) **Action**: to recognize the repeated action; (ii) **Transition**: to identify the transition between the before and after states; (iii) **Frame**: to answer questions about a specific frame from the GIF video. Among them, TGIF-Action and TGIF-Transition are collected under a multiple-choice setting, and TGIF-Frame is

an open-ended video QA task with free-form answers. We adopt batch size 24 and train for 56/20/10 epochs for Action/Transition/Frame, respectively.

MSRVTT-MC [33] and MSRVTT-QA [31] are created based on videos and captions in MSRVTT [32]. MSRVTT-MC is a multiple-choice task with videos as questions, and captions as answers. Each video contains 5 captions, with only one positive match. This setting can be viewed as video-to-text retrieval, hence we simply evaluate the model trained on MSRVTT-Retrieval. MSRVTT-QA contains 243K open-ended questions over 10K videos. We adopt batch size 24 per GPU and training epochs 8.

MSVD-QA [31] consists of 47K open-ended questions over 2K videos, based on video-caption pairs from MSVD [4]. We adopt batch size 24 per GPU and train for 10 epochs.

LSMDC-MC and LSMDC-FiB [29] are built from the LSMDC dataset [26]. Similar to MSRVTT-MC, LSMDC-MC requires the model to select the only positive caption that describes the video from 5 caption candidates, and can be formulated as video-to-text retrieval. LSMDC-FiB replaces a word in the question sentence with the `[BLANK]` token, and requires the model to recover the missing word. We regard LSMDC-FiB as an open-ended Video QA task. In particular, we replace the `[BLANK]` token with `[MASK]` token, and use the MLM prediction head over the representation h_x of the `[MASK]` token to predict the correct answer. We adopt batch size 24 per GPU and train for 10 epochs.

C.3. Video Captioning

For video captioning, we evaluate on MSRVTT [32] and MSVD [5]. **MSRVTT** consists of 10K videos with 20 captions per video, and **MSVD** contains 2K videos, with 40 captions per video. We follow the standard captioning splits to train/evaluate with VIOLETv2. The captioning finetuning is formulated as masked language modeling (MLM) with a causal attention mask so that the current word token only attends to the tokens before it, following SwinBERT [14]. During training, we set the probability of random masking caption tokens to 0.15, the same as what is used in MLM during pre-training. We adopt batch size 24 per GPU and train for 20 epochs. During inference, we generate the captions auto-regressively. At each generation step, a `[MASK]` token is appended to the previously generated tokens, and the model will predict the current tokens based on the learned embedding at the `[MASK]` token position. We perform generation until the model outputs a `[SEP]`, which is defined as the sentence ending token or when it reaches the maximum generation step 50.

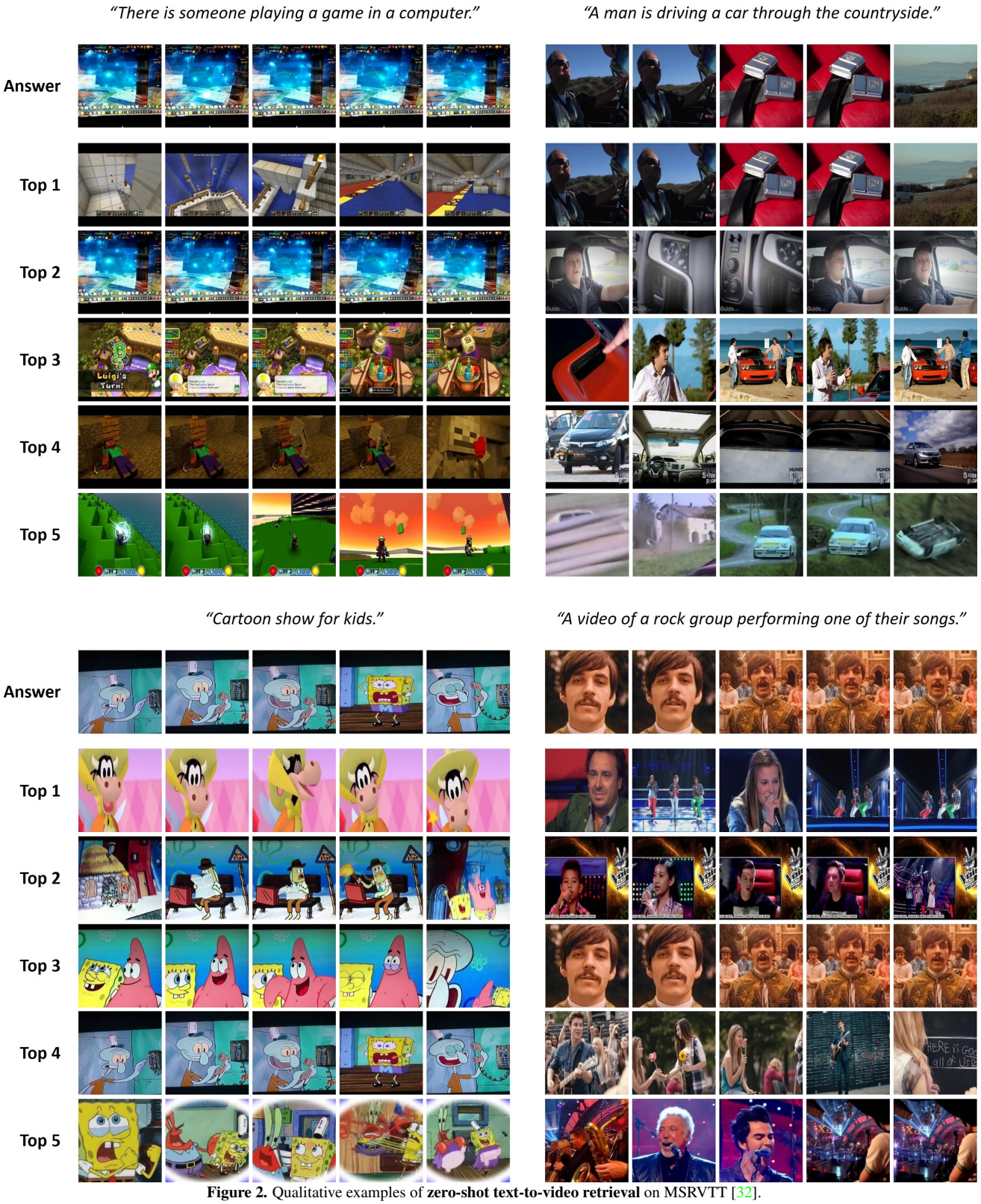


Figure 2. Qualitative examples of **zero-shot text-to-video retrieval** on MSRVTT [32].

Figure 3. Qualitative examples of **zero-shot text-to-video retrieval** on DiDeMo [9].

Figure 4. Qualitative examples of **zero-shot text-to-video retrieval** on LSMDC [26].

References

[1] Microsoft Azure. https://azure.microsoft.com/. 4

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 3, 4

[3] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the "Video" in Video-Language Understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[4] David L. Chen and William B. Dolan. Collecting Highly Parallel Data for Paraphrase Evaluation. In *Annual Meetings of the Association for Computational Linguistics (ACL)*, 2011. 4, 5

[5] David L. Chen and William B. Dolan. Collecting Highly Parallel Data for Paraphrase Evaluation. In *ACL*, 2011. 5

[6] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 3

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. 3

[8] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. VIOLET: End-to-End Video-Language Transformers with Masked Visual-token Modeling. In *arXiv:2111.1268*, 2021. 1

[9] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing Moments in Video with Natural Language. In *International Conference on Computer Vision (ICCV)*, 2017. 4, 7

[10] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[11] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. M3L: Language-based Video Editing via Multi-Modal Multi-Level Transformer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4

[12] 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 *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. 1, 2

[13] Linjie Li, Zhe Gan, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Ce Liu, and Lijuan Wang. Lavender: Unifying video-language understanding as masked language modeling. *arXiv preprint arXiv:2206.07160*, 2022. 4

[14] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. SwinBERT: End-to-End Transformers with Sparse Attention for Video Captioning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[15] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen Clip Models are Efficient Video Learners. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[16] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 3

[17] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video Swin Transformer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[18] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference for Learning Representations (ICLR)*, 2019. 3, 4

[19] 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 *International Conference on Computer Vision (ICCV)*, 2019. 4

[20] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding Language-Image Pretrained Models for General Video Recognition. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 3

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 2021. 3

[23] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. In *International Conference on Machine Learning (ICML)*, 2021. 3

[24] Rene Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision Transformers for Dense Prediction. In *International Conference on Computer Vision (ICCV)*, 2021. 3

[25] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters. In *Knowledge Discovery in Database (KDD)*, 2020. 4

[26] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A Dataset for Movie Description. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 4, 5, 8

[27] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018. 1, 4

[28] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 3

[29] Atousa Torabi, Niket Tandon, and Leonid Sigal. Learning Language-Visual Embedding for Movie Understanding with Natural-Language. In *arXiv:1609.08124*, 2016. 4, 5

[30] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training Data-efficient Image Transformers & Distillation through Attention. In *International Conference on Machine Learning (ICML)*, 2021. 2

[31] 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 Multimedia (ACMMM)*, 2017. 4, 5

[32] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4, 5, 6

[33] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A Joint Sequence Fusion Model for Video Question Answering and Retrieval. In *European Conference on Computer Vision (ECCV)*, 2018. 4, 5

[34] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. MERLOT: Multimodal Neural Script Knowledge Models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 1