LAVENDER: Unifying Video-Language Understanding as Masked Language Modeling

Linjie Li, Zhe Gan, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Ce Liu, Lijuan Wang
Microsoft
{linjli,zhgan,keli,chungching.lin,zliu,ce.liu,lijuanw}@microsoft.com

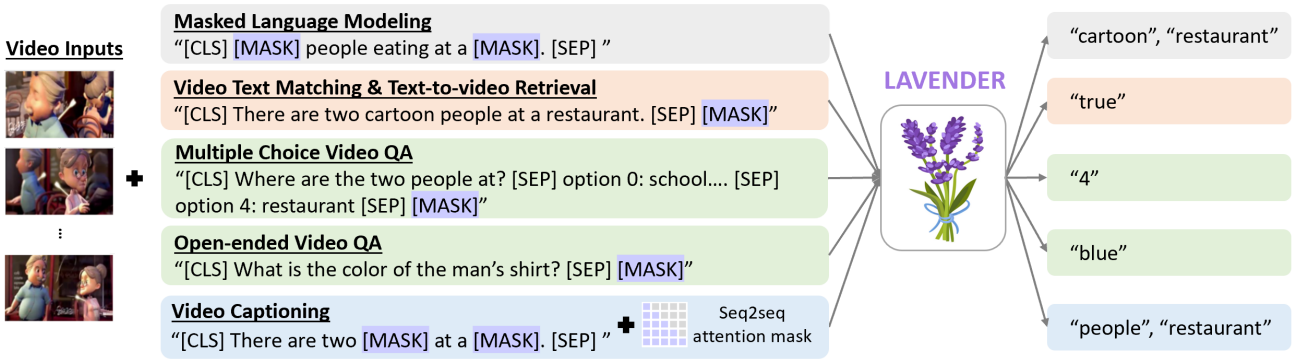


Figure 1. Overview of LAVENDER (**LA**nguage-**V**id**E**o u**NDER**standing) model. LAVENDER unifies both pre-training and downstream finetuning as Masked Language Modeling.

Abstract

Unified vision-language frameworks have greatly advanced in recent years, most of which adopt an encoder-decoder architecture to unify image-text tasks as sequence-to-sequence generation. However, existing video-language (VidL) models still require task-specific designs in model architecture and training objectives for each task. In this work, we explore a unified VidL framework LAVENDER*, where Masked Language Modeling [13] (MLM) is used as the common interface for all pre-training and downstream tasks. Such unification leads to a simplified model architecture, where only a lightweight MLM head, instead of a decoder with much more parameters, is needed on top of the multimodal encoder. Surprisingly, experimental results show that this unified framework achieves competitive performance on 14 VidL benchmarks, covering video question answering, text-to-video retrieval and video captioning. Extensive analyses further demonstrate* LAVENDER *can (i) seamlessly support all downstream tasks with just a single set of parameter values when multi-task finetuned; (ii) generalize to various downstream tasks with limited training samples; and (iii) enable zero-shot evaluation on video question answering tasks. Code is available at https://github.com/microsoft/LAVENDER.*

1. Introduction

Large-scale transformer-based pre-training is now the *de facto* practice for NLP and vision-language research [13, 25, 37, 49, 50]. Together with the great success of image-text pre-training [10, 35, 40, 59], video-language (VidL) pre-training [29, 33, 58, 80, 83] has also received an increasing amount of attention. By pre-training an end-to-end multimodal transformer on a large number of video-text pairs, state-of-the-art performance has been achieved across a wide range of VidL tasks, including video question answering (QA) [24, 68], text-to-video retrieval [22, 51], and video captioning [64, 71]. These advances are encouraging; however, on the other hand, all existing VidL works require designing task-specific heads on top of the transformer encoder for each pre-training or downstream task. For example, during pre-training, separate Masked Language Modeling [13] (MLM) and Video Text Matching (VTM) heads are used, while a new, separately parameterized head needs to be added for each downstream adaptation. Furthermore, due to the particular nature of different tasks, they are typically modeled using different training objectives. For example, multiple-choice video QA is formulated as a classification problem, while video captioning is inherently a generation task. A natural but challenging question arises: *can we have a unified architecture that supports all the popular VidL tasks simultaneously without introducing task-specific heads?*

To answer this, we present LAVENDER, a unified VidL

	TGIF			MSRVTT		LSMDC		MSVD	Captioning	Retrieval
	Action	Transition	Frame	MC	QA	MC	FiB	QA	MSVD	DiDeMo
Published	94.0	96.2	69.5	90.9	43.1	81.7	52.9	46.3	120.6	65.1
SOTA	[80]	[80]	[80]	[80]	[80]	[80]	[80]	[73]	[36]	[5]
LAVENDER	94.8	98.7	73.5	97.2	45.0	85.9	57.1	55.6	150.3	72.4
Δ	0.8↑	2.5↑	4.0↑	6.3↑	1.9↑	4.2↑	4.2↑	9.3↑	29.7↑	7.3↑

Table 1. New state-of-the-art performance with LAVENDER (14M+16M pre-train, single-task finetune) across 10 VidL tasks. Accuracy, average(R1, R5, R10) and CIDEr scores are reported for video QA, retrieval and captioning.

framework where all pre-training and downstream tasks are formulated as simple MLM. As shown in Figure 1, we use two pre-training tasks: MLM and VTM. However, for VTM, instead of adding a head on top of the output of the commonly used `[CLS]` token, as used in all existing works, we propose to append the same `[MASK]` token that is used for MLM at the end of the video-text input, and use the same MLM head to predict whether the video-text pair matches or not. Note that VTM is typically formulated as a binary classification problem; here, we simply treat the output of `true` or `false` from VTM as natural language tokens directly predicted from the whole vocabulary, so that the same set of parameters can be used for both MLM and VTM.

During downstream adaptation, instead of following standard practice in VidL literature to replace the MLM head in pre-training with new heads, we use the same MLM head used in pre-training for all downstream tasks. Specifically,

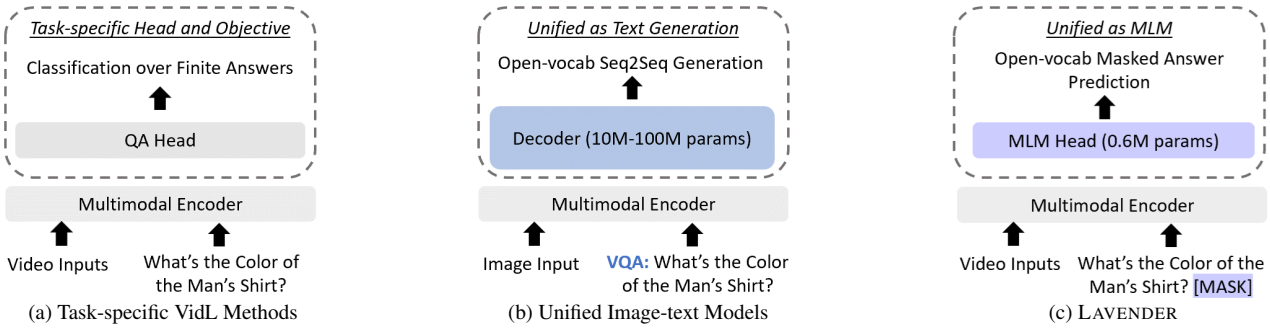
- For text-to-video retrieval, we train the model in the same way as in the VTM pre-training task. During inference, for each text query, we concatenate it with each candidate video, and calculate the corresponding probability of the `[MASK]` token being predicted as `true`, and then rank all candidate videos based on that score.
- For multiple-choice video QA, we concatenate the question and each answer candidate sequentially, and add a `[MASK]` token at the end of the sequence, and use the same MLM head to predict the answer as "`n`" (assuming the ground-truth choice is the `n`-th answer).
- For open-ended video QA, since most of the ground-truth answers in our tested datasets only contain one word, we simply append a `[MASK]` token to the end of the video-question input, and let the model predict the answer from the whole vocabulary.
- For video captioning, during training, we mask a certain percentage of the tokens, and then predict the masked tokens using a seq2seq attention mask [35, 81]. During inference, the full caption is auto-regressively predicted, by inserting `[MASK]` tokens one at a time.

LAVENDER is inspired by VL-T5 [12], UniTAB [76] and OFA [63] that aim to provide a unified pre-training framework for image-text tasks. However, LAVENDER adopt an encoder-only model and an additional lightweight MLM head on top of it, while a heavy transformer decoder is needed in [12, 63, 76]. By unifying all VidL tasks as MLM, LAVENDER can seamlessly adapt to different VidL tasks, meanwhile (i) support different VidL tasks with a single set of parameter values when multi-task finetuned; (ii) generalize to test data under few-shot finetuning; and (iii) enable zero-shot inference on video question answering. Surprisingly, by using this simple generative approach, we outperform previously published state-of-the-arts on 10 out of 14 downstream tasks (Table 1), even when pre-trained with much fewer data (Section 4.5).

2. Related Work

Video-Language Pre-training. Branching out from large-scale image-text pre-training [10, 59], researchers have been leveraging large-scale multimodal data [3, 9, 27, 46, 55, 79, 80] to build pre-trained video-language (VidL) models [1, 16, 18, 28, 42, 48, 52, 69, 70, 75] for a wide range of generative [31, 71, 82] and discriminative [24, 30, 44] tasks. Prominent examples include VideoBERT [58], HERO [33], ActBERT [83], ClipBERT [29] and MERLOT [80]. Popular pre-training tasks include Masked Language Modeling (MLM) [57], Video Text Matching (VTM) [29], frame order modeling [33, 80] and masked visual modeling [15, 33]. Although achieving strong performance, existing methods all require task-specific architectures or objectives for different downstream tasks. For example, text-to-video retrieval [22, 51] is modeled as binary classification [29] or via contrastive learning [17, 45]; video question answering [24, 68] is often formulated as multi-class classification with a pre-defined answer set [77, 80]; and video captioning can be tackled via MLM with a multi-layer perceptron [36] or prefix language modeling with a text decoder [54].

Unified Frameworks for Multimodal Understanding. There have been attempts in building an omnipotent model that can simultaneously handle different tasks with a unified architecture, which can be broadly categorized into two directions. The first is to insert expert-designed task-specific heads for each downstream task [21, 23, 34, 56]. These task-specific output layers not only require expert knowledge, but are also unlikely to generalize to new tasks. For example, when a new question answering task comes in, a new fully-connected layer with output dimension of answer vocabulary size is required. The second direction is to unify the input-output format of different downstream tasks [12, 63, 66, 76]. With a unified vocabulary, different downstream tasks (*e.g.*,



(a) Task-specific VidL Methods (b) Unified Image-text Models (c) LAVENDER

Figure 2. Illustration of the **differences between LAVENDER and existing methods** with image/video QA as an example. Unlike task-specific designs in existing VidL methods, LAVENDER unifies all tasks as MLM (Figure 1). We adopt an encoder-only architecture, with a lightweight MLM head, instead of the heavy decoder in unified image-text models (*e.g.*, VL-T5 [12] with task-specific prefix in text input).

VQA [20] and image captioning [9]) can be formulated as the sequence-to-sequence generation with the shared encoder-decoder architecture [2, 63, 74, 76].

Our work aims to provide a unified framework for VidL understanding, in contrast to task-specific architectures and objectives used in existing VidL models (Figure 2c vs. Figure 2a). LAVENDER differs from the previous unified models in that all pre-training and downstream tasks are unified as MLM, and a simple encoder-only architecture with a lightweight MLM head is used, instead of sequence-to-sequence modeling as in [2, 12, 66, 74] that also requires a heavy transformer decoder (Figure 2c vs. Figure 2b).

3. LAVENDER

3.1. Model Architecture

Given a pair of text sentence $\{\boldsymbol{w}_n\}|_{n=1}^{N}$ and a video $\{\boldsymbol{v}_t\}|_{t=1}^{T}$, we first encode them separately via unimodal encoders (*i.e.*, vision encoder and text encoder) to generate unimodal features. Here, N is the number of tokens in a sentence and T is the number of frames sampled from the input video. We follow previous works [29, 80] to only sparsely sample a few frames to ease the computational burden. A multimodal fusion encoder (dubbed as fusion encoder) projects textual features and visual features into a shared embedding space to learn cross-modal representations. As LAVENDER unifies both pre-training and downstream tasks as Masked Language Modeling (MLM), the same MLM head is used to generate the final outputs from the cross-modal representations, across different tasks. Next, we explain each component in detail.

Vision Encoder. Inspired by the success of vision transformers in modeling spatial details in images [14, 38], different transformer architectures [4, 39, 72] have been proposed to model the long-range temporal modeling in videos, achieving promising results on action recognition [26]. Recent video-language works [15, 36] have started to embrace the success of video transformers, demonstrating stronger performance than encoding each video frame independently [29]. In our work, we adopt Video Swin Transformer [39] (VidSwin) as the vision encoder to encode the raw video frame inputs as a sequence of visual features. Given T input video frames $\{\boldsymbol{v}_t\}|_{t=1}^{T}$ of size $H \times W \times 3$, we first split each frame into non-overlapping patches of size $h \times w$. VidSwin additionally enforces temporal downsampling of size 2 as a preprocessing step. To allow LAVENDER to utilize both video-text and image-text data for pre-training, we remove this temporal downsampling. As a result, we can extract a sequence of visual features of size $T \times \frac{H}{h} \times \frac{W}{w}$ from the last encoder block of VidSwin. Each feature is of size $8C$ (C is the channel dimension), which is projected to the same dimensional space as text features via a fully-connected layer. We follow [15] to add learnable positional embedding layers along both spatial and temporal dimensions. The resulting visual features are used as input to the fusion encoder to learn cross-modal representations.

Text Encoder. The input text sentence is first tokenized into the sequence of word tokens $\{\boldsymbol{w}_n\}|_{n=1}^{N}$, following [67]. Two special tokens `[CLS]` and `[SEP]` are inserted at the beginning and the end of the token sequence. We follow previous works [15, 29, 80] to adopt a lightweight word embedding layer [13] as the text encoder. The high-dimensional text embeddings are concatenated with visual features and then fed into the fusion encoder.

Multimodal Fusion Encoder. The fusion encoder is a 12-layer, 768-dimensional Transformer [61], mirroring the BERT-base architecture [13]. To compute the cross-modal representations, the unimodal features from vision and text encoders are fused together via self-attention operations.

3.2. Our Unified Framework

Now, we introduce how to train LAVENDER in a unified way, with the same MLM objective and the shared vocabulary for both pre-training and downstream finetuning.

Video-language Pre-training. We adopt two objectives to pre-train LAVENDER. The first is **Masked Language Modeling** (MLM), which is directly adopted from language model pre-training [13, 37]. In MLM, we randomly replace 15% of word tokens with a `[MASK]` token, a random word,

or the same word. The goal is to reconstruct the correct tokens based on the corresponding hidden representations from the output of the fusion encoder at the masked position. A multi-layer perceptron (MLP) with output dimension as `vocab_size`[1] projects these hidden representations into the discrete word token space. Cross-entropy loss is used to supervise the model training. The second is **Video Text Matching**, but reformatted as MLM (VTM as MLM). Specifically, we append a `[MASK]` token to the textual sentence to mimic the masked textual inputs in MLM. At each training step, we randomly replace the corresponding text for a given video with a text description from a different video in the same batch. At the masked position, LAVENDER reuses the exact same MLP used in MLM to make a prediction. Although the ground-truth label is restricted to two tokens (*i.e.*, `true` (`false`) for a positive (negative) video-text pair), but the model predictions are made across all vocabularies.

Downstream Adaptation. As shown in Figure 1, we can readily apply the pre-trained LAVENDER to 4 types of downstream tasks, including text-to-video retrieval, multiple-choice / open-ended video question answering, and video captioning. For each task, we transform the text input by inserting or replacing existing tokens with `[MASK]` tokens, so that all tasks can be supervised with cross-entropy loss, and the final predictions are made based on the word token predicted at the masked position. Here, we explain in detail how to construct the masked textual inputs and generate model predictions for each downstream task.

For **text-to-video retrieval**, similar to VTM during pre-training, we insert a `[MASK]` token at the end of the text input. During training, we treat corresponding video-text pairs as positives (with ground-truth label `true`) and all other pairwise combinations constructed by replacing the ground-truth text with a randomly sampled one as negatives (with ground-truth label `false`). During inference, given a textual query, we rank the videos according to the model confidence of predicting `true` at the masked position. For **multiple-choice video QA**, we concatenate each answer choice ($\texttt{A}_\texttt{n}$) sequentially to the question (`Q`) with a `[SEP]` token in between. A `[MASK]` token is then added at the end, to allow the model to make a prediction of the correct index for the ground-truth answer choice. For example, for a question and 5 answer choices, we take $\texttt{Q+[SEP]+A}_0\texttt{+[SEP]+...+A}_4\texttt{+[MASK]}$ as the text input. If $\texttt{A}_\texttt{n}$ is the correct answer, the ground-truth label for the masked token is `n`. Through the MLM head, the model makes a prediction at the masked position over the whole vocabulary. During inference, to ensure a valid answer, we take the most probable predictions over all answer indices (*e.g.*, {`0,1,2,3,4`}). Conventional methods [15, 33] concatenate a single answer with the question at a time, and model as multi-class classification. Intuitively, concatenating all answers may better mimic how humans tackle a MC question (e.g., we often read through all options to conclude an answer).[2] For **open-ended video QA**, we similarly inject `[MASK]` tokens after the question. For simplicity, we only add one `[MASK]` token.[3] We then tokenize the ground-truth answers as the ground-truth label for masked prediction. If the tokenized answer is longer than 1 word, we simply ignore it during training, and regard it as a wrong prediction during inference. For **video captioning**, we use a causal self-attention mask where the caption token can only attend to the existing output tokens, which simulates a uni-directional seq2seq generation process, following [36]. During training, we randomly "mask" some words with `[MASK]` token and apply the MLM objective. During inference, the caption is generated in an auto-regressive manner. At each generation step, the model sees the entire video input and previously generated tokens, plus a `[MASK]` token, at which the model makes a prediction for the current token.

[1]The `vocab_size` is 30,522, following `bert-base-uncased`.

4. Experiments

In this section, we first describe our experimental settings (4.1), and show the superiority of LAVENDER over comparable task-specific baselines under both single-task (4.2) and multi-task (4.3) finetuning settings. We then show that our model can better generalize to testing data under few-shot finetuning and has strong zero-shot capability on video question answering benchmarks (4.4). Lastly, we compare LAVENDER with prior arts and show we outperform on 11 out of 14 benchmarks, even when pre-trained with much fewer data (4.5).

4.1. Experimental Settings

Pre-training Data. In our default setting, we follow [3] to aggregate video-text pairs in WebVid2.5M [3] and image-text pairs in CC3M [55] to pre-train LAVENDER. As a scale-up recipe, we additionally crawl 11.9M video-text pairs from the web, following the same procedure in [3]. We similarly scale up image-text pairs by assembling COCO [9], Visual Genome [27], SBU Captions [47], CC12M [7] and CC3M. Combining these video-text and image-text datasets together results in 14M videos + 16M images. Unless otherwise specified, all results reported in this section as LAVENDER are pre-trained under the default setting with 2.5M videos + 3M images. In Section 4.5, we show that scaling up our pre-training data further improves model performance.

Downstream Tasks. We evaluate LAVENDER on 14 video-language benchmarks over popular VidL tasks, including text-to-video retrieval, video question answering (QA) in

[2]Quantitatively, we observe concatenating a single answer results in some performance drop (*e.g.*, 98.7 → 94.0 on TGIF-Transition).

[3]Note that we can optionally insert multiple `[MASK]` tokens to allow answer predictions with variant lengths. However, over 95% of the questions in the evaluation datasets considered can be answered with a single word, as shown in Appendix.

VidL Pre-training	Task-specific designs	Finetune setting	#Params	#	Meta Ave.	TGIF Action	MSVD QA	DiDeMo Ret.	MSRVTT Cap.
-	-	ST	4(P+H)	1	45.5	93.5	40.8	0.0[4]	47.7
		MT	P+H	2	58.5	95.9	47.4	41.2	50.0
	Head	ST	4(P+H)	3	40.1	31.9	44.2	36.7	47.4
		MT	P+4H	4	55.6	94.1	44.6	35.4	48.3
VTM+MLM	Head	ST	4(P+H)	5	64.0	94.5	46.7	59.0	55.7
		MT	P+4H	6	62.4	95.5	47.7	53.0	53.3
VTM (as MLM)+MLM	-	ST	4(P+H)	7	**68.9**	95.8	**54.4**	**68.2**	57.3
	-	MT	P+H	8	68.3	**96.5**	53.5	65.8	**57.4**
	Task Prompt			9	67.9	96.2	53.4	65.6	56.4
	Task Token			10	67.9	**96.5**	53.6	64.9	56.7

Table 2. **Comparison to task-specific baseline** under single-task (ST) and multi-task (MT) finetuning, with or without video-language (VidL) pre-training. We report accuracy for QA, average (R1, R5, R10) for retrieval (Ret.) and CIDEr score for captioning (Cap.). Meta-Ave. is the average across all scores. P and H denote the total parameter count in the backbone of LAVENDER (vision encoder + text encoder + fusion encoder) and top output layer. Note that the baseline with task-specific heads (L3-6) is LAVENDER-TS, introduced in Section 4.2.

both multiple-choice (MC) and open-ended (OE) settings and video captioning. We briefly list the evaluation datasets for each task type below.

- Text-to-video Retrieval: MSRVTT [71], DiDeMo [22], MSVD [8] and LSMDC [51];
- MC Video QA: TGIF-Action, TGIF-Transition [24], MSRVTT-MC [78] and LSMDC-MC [60];
- OE Video QA: TGIF-Frame [24], MSRVTT-QA, MSVD-QA [68] and LSMDC-FiB [44];
- Video Captioning: MSRVTT [71] and MSVD [8].

Implementation Details. We initialize our Vision Encoder with VideoSwin-Base [39], pre-trained on Kinetics-600 [26]. Text Encoder and Multimodal Fusion Encoder are initialized from pre-trained BERT-Base [13]. LAVENDER is end-to-end trained for both pre-training and downstream finetuning. For default setting with 2.5M videos + 3M images, we pre-train LAVENDER for 10 epochs on 32 NVIDIA V100 GPUs, which takes about 2 days. The scale-up pre-training takes about 10 days on 64 NVIDIA V100 GPUs. More implementation details can be found in Appendix.

4.2. Comparison to Task-specific Baseline

To make a fair comparison to task-specific methods, we train a task-specific version of LAVENDER (denoted as LAVENDER-TS). We replace the shared Masked Language Modeling (MLM) head in LAVENDER with task-specific heads and adopt task-specific objectives. For **text-to-video retrieval** (and similarly for video text matching during pre-training), a multi-layer perceptron (MLP) with output dimension 1 is applied over the global video-language representation of the `[CLS]` token and binary cross-entropy loss is adopted to supervise the model training. For **multiple-choice video question answering** (QA), we concatenate questions with all answer candidates to form the input text, similar to what was described in Section 3.2, but without the added `[MASK]` token. A task-specific MLP with output dimension as the number of answer choices is applied over the representation of `[CLS]` token and cross-entropy loss is used to train a classifier over all answer indices (*e.g.*, $\{0, 1, 2, 3, 4\}$ with 5 answer choices). For **open-ended video QA**, we follow the common practice [15, 80] to build a finite set of answer vocabularies covering the most common answers in the training split of each dataset. Similarly, a MLP with output dimension as the number of answers is added and cross-entropy loss is used to train a classifier over all answers. For **video captioning**, we simply adopt the same training strategy with MLM head as in our unified model.

Table 2 compares LAVENDER to the task-specific baseline LAVENDER-TS under different settings, on four representative benchmarks. For easier comparisons, we measure the average model performance with Meta-Ave, the average across scores over all evaluation tasks. Here, we focus our discussion on results under single-task finetuning, and defer the analysis on multi-task finetuning to Section 4.3. Without video-language (VidL) pre-training, LAVENDER-TS with task-specific heads (L3) outperforms our unified model (L1) on MSVD-QA and DiDeMo Retrieval. Captioning performance are similar as the same MLM head and finetuning strategy are applied to both models. On TGIF-Action, we empirically find the training of LAVENDER-TS struggles to converge, leading to a low Meta-Ave.

We also compare the two models under VidL pre-training. We follow [29] to pre-train task-specific LAVENDER-TS with MLM and the standard Video Text Matching (VTM) task, which is modeled as binary classification with an additional MLP layer. In comparison, the unified model LAVENDER is pre-trained with MLM and VTM as MLM, with the shared MLM head. The unified VidL pre-training (L7) significantly enhances the performance of LAVENDER, with a gain of +23.4 on Meta-Ave over without pre-training (L1). Comparing both models under VidL pre-training, we also observe that LAVENDER (L7) outperforms LAVENDER-TS (L5) by a

[4]We empirically observe that the finetuning of DiDeMo retrieval with LAVENDER without pre-training did not converge. This result indicates that in order to model retrieval task as MLM, where the answer is limited to two words (`true` or `false`) instead of 30,522 words, the model has to learn from more data (*e.g.*, pre-training or multi-task finetuning).

Finetune Method	# Params	Meta Ave.	TGIF			MSRVTT				LSMDC			MSVD			DiDeMo
			Act.	Trans.	Frame	MC	QA	Ret	Cap	MC	FiB	Ret	QA	Ret	Cap	Ret
ST	14P	73.9	95.8	**99.1**	**72.2**	**96.6**	**44.2**	**58.9**	57.3	84.5	**56.9**	**39.8**	54.4	67.6	139.4	**68.2**
MT (all-in-one)	P	73.4	95.8	98.0	70.7	93.9	44.1	56.3	57.1	85.3	56.5	39.4	53.4	69.2	141.1	66.1
MT (best)	14P	73.8	95.8	98.3	71.6	94.3	**44.2**	56.4	57.2	**86.0**	56.7	39.4	**55.4**	69.3	141.6	66.5
MT → ST	14P	**74.2**	**96.6**	98.5	71.2	96.0	44.1	58.8	**58.0**	85.3	**56.9**	**39.8**	53.5	**69.7**	**142.9**	67.7
MT (all-in-one) TS	>P	69.2	93.8	97.2	65.4	92.2	41.7	52.7	54.2	83.0	49.5	34.7	49.2	65.6	133.7	56.5

Table 3. **Multi-task Finetuning** under VidL pre-training. Accuracy, average (R1, R5, R10) and CIDEr score are used as evaluation metrics for video QA, retrieval and captioning tasks. Meta-Ave. is the average score across all evaluation datasets. P denotes the total parameter count in LAVENDER. MT (all-in-one) TS is LAVENDER-TS trained under all-in-one setting, all others are based on LAVENDER.

notable margin across all 4 tasks (+4.9 on Meta-Ave).

4.3. Multi-task Finetuning

In this section, we aim to answer the important question raised in Section 1: *can we have a unified architecture that supports all downstream tasks simultaneously without introducing task-specific heads?* We first compare LAVENDER with several multi-task learning baselines with different task-specific designs in Table 2 and report results under the *extreme* multi-task setting - a single model that can support all 14 datasets (all-in-one) in Table 3.

Comparison to task-specific baseline. We begin our comparison with the most common multi-task baseline in literature [34, 41] - adopting a task-specific head and objective for each task, while sharing the backbone. This is equivalent to finetuning the task-specific model LAVENDER-TS under multi-task setting. We compare the two models in Table 2 and summarize our findings below:

- **Single-task vs. Multi-task:** Without video-language (VidL) pre-training (L1-4), we find that multi-task training greatly improves model performance in both cases, as it can take advantages of additional and diverse supervision. Multi-task finetuning can also be regarded as a way of pre-training. Table 2 results suggest that pre-training only (L5/L7) on larger-scale data gives better performance than multi-task only (L4/L2), with a gain of +8.8 for LAVENDER-TS and +10.4 for LAVENDER, respectively. Combining multi-task finetuning with VidL pre-training renders slight performance drop (L5-8) for both models.
- **Single shared head vs. Task-specific heads:** LAVENDER (single shared head) outperforms LAVENDER-TS (task-specific heads), with (L8 vs. L6, +5.9) or without (L2 vs. L4, +2.9) VidL pre-training. LAVENDER also saves more parameters (approximately 3H) from the additional 3 task-specific heads in LAVENDER-TS under multi-task finetuning. In addition, the performance drop of multi-task finetuning from single-task finetuning is less severe with the shared MLM head (-0.6 for LAVENDER vs. -1.6 for LAVENDER-TS) under VidL pre-training.

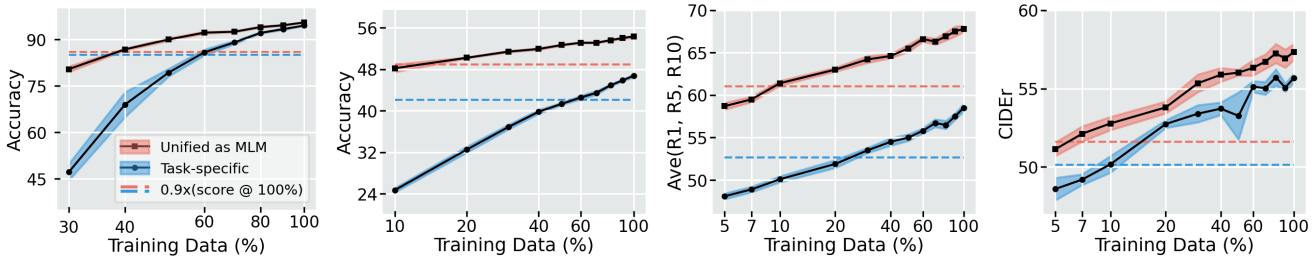
Multi-task Variants with LAVENDER. We also explore different multi-task variants with our unified framework LAVENDER in Table 2: (i) L8: the vanilla version without any task-specific design; (ii) L9: with human-readable task-specific prompts (*e.g.*, "*is the video-text paired, true or false?*" for video text matching), which has shown promising results for language understanding [53]; and (iii) L10: with learnable task-specific tokens (*e.g.*, a special token `[VTM]` for video text matching), which is in analogy to the task prefixes in [12]. Different from observations in [12,53], both task-specific prompts and tokens do not show a clear advantage over the vanilla version. We conjecture the differences may be due to the weaker text encoder and the less diverse text prompts, which we leave as interesting directions for future study. Based on the above analyses, we simply extend the vanilla multi-task finetuning method from 4 to all 14 VidL benchmarks considered.

All-in-One. In Table 3, we finally attempt to answer the question with one model that can conquer all 14 downstream tasks simultaneously. We first establish the baseline performance by training single-task (ST) models with LAVENDER. We then report multi-task results with (i) a single set of parameters for all tasks (MT (all-in-one)); (ii) the best-performing checkpoint for each task (MT (best)); and (iii) with multi-task finetuning as 2nd stage pre-training and then finetune on each task (MT→ST), from the learned weights with MT (all-in-one). As the results show, MT→ST achieves slightly better Meta-Ave across all settings. Surprisingly, the all-in-one model is very competitive, with only -0.5 performance drop on Meta-Ave, when compared with ST models. Under all-in-one setting, our unified method consistently outperform task-specific baseline with a gain of +4.2 on Meta-Ave. We explore additional MT settings in Appendix, and find that all-in-one empirically strikes a balance between sophisticated heuristic designs of multi-task setting and good model performance.

4.4. Few-shot and Zero-shot Evaluation

Next, we showcase two capabilities enabled by LAVENDER over task-specific baseline.

Few-Shot Generalizability. We first study how LAVENDER can generalize to testing data with limited training examples. Figure 3 compares the results of our unified LAVENDER (red line), against the task-specific baseline LAVENDER-TS with different heads and finetuning objectives (blue line) on 4 representative benchmarks. The two dotted lines denote 90% of model performance when trained with all the training data



(a) TGIF-Action (b) MSVD-QA (c) DiDeMo Retrieval (d) MSRVTT Captioning

Figure 3. **Few-shot Evaluation** under VidL Pre-training. Each experiment are repeated 5 times with different random seeds. The shaded areas highlight the standard error. Percentage of training data needed to achieve 90% of the full model performance: (a) 40%, (b) 10%, (c) 10%, (d) 6% for LAVENDER (unified as MLM, red) and (a) 60%, (b) 60%, (c) 25%, (d) 10% for task-specific baseline LAVENDER-TS (blue).

Method	# pre-train video/images	TGIF			MSRVTT		LSMDC		MSVD
		Act.	Trans.	Frame	MC	QA	MC	FiB	QA
JustAsk [73]	69M / -	-	-	-	-	2.9	-	-	7.5
MERLOT RESERVE [79]	1B/ -	-	-	-	-	5.8	-	31.0	-
BLIP [32]	- / 129M	-	-	-	-	19.2	-	-	35.2
Flamingo [2]	2.1B / 27M	-	-	-	-	19.2	-	-	35.2
FrozenBiLM [74]	- / 10M	-	-	41.9	-	16.9	-	51.5	33.8
All-in-one [62]	283M / -	-	-	-	80.3	-	56.3	-	-
LAVENDER-TS	2.5M / 3M	48.5	47.9	0.0	84.6	0.0	66.9	0.0	0.0
LAVENDER	2.5M / 3M	52.6	**54.1**	16.7	86.7	**4.5**	73.8	34.2	**11.6**
	14M / 16M	**55.1**	53.8	**19.6**	**87.2**	2.7	**73.9**	**36.7**	9.2

Table 4. **Zero-shot Evaluation on Video QA** (top-1 accuracy). Models are evaluated directly after pre-training. BLIP [32] is additionally supervised with VQA v2 [20], and MERLOT RESERVE [79] is pre-trained with additional audio modality and uses GPT-3 [6] to reword questions into masked statements. Flamingo [2] and FrozenBiLM [74] leverage large language models with more than 8x more parameters than the BERT-Base model in LAVENDER.

for each model. Note that both models learn from the exact same amount of data, that is, ∼5M images/videos during pre-training and the same percentage of training data during single-task finetuning. LAVENDER shows a clear advantage of easily achieving 90% of the full model performance, with much less training data. Specifically, approximately 40%, 10%, 10% and 6% of training data is needed for LAVENDER on TGIF-Action, MSVD-QA, DiDeMo-Retrieval and MSRVTT-Captioning, while LAVENDER-TS requires 60%, 60%, 25% and 10%, respectively.

Zero-Shot Evaluation on Video QA. Table 4 compares zero-shot (ZS) performance of LAVENDER with task-specific baseline LAVENDER-TS on 8 video QA benchmarks. Since the model has neither learned to perform the multiple-choice QA task nor seen similar data during pre-training, we transform multiple-choice QA as Video Text Matching (VTM) for better ZS performance. Specifically, we let LAVENDER to predict `true` or `false` via MLM head, given a video-question-answer input, and we rank the probability of model prediction as `true` across all answer choices. Similarly, for LAVENDER-TS with binary classification head for VTM, we simply rank the probability of model prediction as "matched". With the same pre-training data, LAVENDER evidently outperforms LAVENDER-TS on all multiple-choice QA benchmarks. On open-ended QA tasks, LAVENDER can be applied seamlessly, thanks to the shared MLM head. However, the randomly initialized task-specific heads of LAVENDER-TS give meaningless ZS predictions.

We also compare LAVENDER against previous methods. Without the help of additional audio modality in [79] or supervision signals in [32], LAVENDER achieves competitive ZS performance, even when pre-trained with much less data (5.5M vs. >69M). When scaling up the pre-training data by roughly 5 times, we observe notable performance improvements on most QA benchmarks. The performance drop on a few datasets may be due to the inclusion of more noisy data when scaling up. It is worth noting that advanced techniques in concurrent studies [2, 65, 74], such as more powerful language models, larger-scale pre-training and enhanced pretraining schema with in-context pairs are orthogonal to our study and which we believe can be leveraged to further improve LAVENDER performance in future work.

4.5. Comparison to Prior Arts

In this section, we compare LAVENDER with prior arts, which are mostly designed to tackle a single type of video-language (VidL) task. Note that LAVENDER performance are reported on the best finetuning setting for each task, we include more detailed results in Appendix.

Table 5 summarizes results of LAVENDER on video question answering (QA) and video captioning. For **video QA**, LAVENDER achieves significant gains over existing VidL pre-trained models on 7 out 8 video QA benchmarks con-

Method	# Pretrain videos/images	# Params in Backbone	TGIF			MSRVTT		LSMDC		MSVD	Captioning	
			Act.	Trans.	Frame	MC	QA	MC	FiB	QA	MSRVTT	MSVD
ClipBERT [29]	- / 200K	137M	82.8	87.8	60.3	88.2	37.4	-	-	-	-	-
JustAsk [73]	69M / -	166M	-	-	-	-	41.5	-	-	46.3	-	-
MERLOT [80]	180M / -	219M	94.0	96.2	69.5	90.9	43.1	81.7	52.9	-	-	-
VIOLET [15]	183M / 3M	198M	92.5	95.7	68.9	91.9	43.9	82.8	53.7	47.9	-	-
All-in-one [62]	283M / -	110M	95.5	94.7	66.3	92.3	**46.8**	84.4	-	48.3	-	-
SwinBERT [36]	- / -	198M	-	-	-	-	-	-	-	-	53.8	120.6
MV-GPT [54]	53M / -	314M	-	-	-	-	41.7	-	-	-	60.0	-
LAVENDER	2.5M / 3M	198M	**96.6**	**99.1**	72.2	96.6	44.2	86.0	56.9	55.4	58.0	142.9
	14M / 16M		96.3	98.7	**73.5**	**97.4**	45.0	**87.0**	**57.1**	**56.6**	**60.1**	**150.7**

Table 5. Comparison with SOTA on **video QA** (accuracy) and **captioning** (CIDEr).

Method	# Pretrain videos/images	# Params in Backbone	Text-to-Video Retrieval			
			MSRVTT	DiDeMo	MSVD	LSMDC
ClipBERT [29]	- / 200K	137M	22.0 / 46.8 / 59.9	20.4 / 48.0 / 60.8	-	-
Frozen [3]	2.5M / 3.2M	232M	32.5 / 61.5 / 71.2	31.0 / 59.8 / 72.4	45.6 / 79.8 / 88.2	15.0 / 30.8 / 39.8
VIOLET [15]	183M / 3M	198M	34.5 / 63.0 / 73.4	32.6 / 62.8 / 74.7	-	16.1 / 36.6 / 41.2
All-in-one [62]	103M / -	110M	37.9 / 68.1 / 77.1	32.7 / 61.4 / 73.5	-	-
BridgeFormer [19]	- / 400M	∼149M	44.9 / 71.9 / 80.3	-	**54.4 / 82.8** / 89.4	21.8 / 41.1 / 50.6
QB-Norm [5]	- / 400M	∼149M	47.2 / 73.0 / 83.0	43.3 / 71.4 / 80.8	47.6 / 77.6 / 86.1	22.4 / 40.1 / 49.5
CAMoE [11]	- / 400M	∼149M	**47.3 / 74.2 / 84.5**	43.8 / 71.4 / 79.9	49.8 / 79.2 / 87.0	25.9 / 46.1 / 53.7
LAVENDER	2.5M / 3M	198M	37.8 / 63.8 / 75.0	47.4 / 74.7 / 82.4	46.3 / 76.9 / 86.0	22.2 / 43.8 / 53.5
	14M / 16M		40.7 / 66.9 / 77.6	**53.4 / 78.6 / 85.3**	50.1 / 79.6 / 87.2	**26.1 / 46.4 / 57.3**

Table 6. Comparison with SOTA on **text-to-video-retrieval** (R1/5/10). CAMoE [11] assumes the model can see all queries during testing.

sidered. On MSRVTT-QA, LAVENDER is only 1.8 points behind All-in-one [62] pre-trained with 283M videos, which is 9 times more than ours (30M). It is worth mentioning that VIOLET [15] adopts the same model architecture and fine-tuning objectives as our task-specific baseline LAVENDER-TS. Even when scaling up the VidL pre-training to 186M videos+images, the task-specific model VIOLET still underperforms LAVENDER, which further demonstrates the advantages of our unified framework. Due to computational constraint, we leave even larger-scale pre-training with >100M data as future study. For **video captioning**, LAVENDER achieves the new state-of-the-arts on both datasets. Note that MV-GPT [54] is pre-trained for multi-modal video captioning, where the auto-transcribed text from audio is used as additional input. With video-only inputs, LAVENDER is able to achieve comparable performance. Furthermore, we also include comparison in number of model parameters, LAVENDER is of comparable model size, but requires less data to achieve better performance.

Table 6 presents the comparison on **text-to-video retrieval**. The most competitive methods [5, 11, 19] on text-to-video retrieval are based on CLIP [49] pre-trained on 400M images. However, with much fewer pre-training data, LAVENDER can still perform competitively on all 4 benchmarks, especially when compared to non-CLIP pre-trained methods [3, 15, 62]. Notably, on DiDeMo and LSMDC, LAVENDER surpasses all baseline methods in Table 6. We hypothesize that the fusion encoder in LAVENDER is more effective in modeling interactions between video and long paragraph query (*i.e.*, DiDeMo) or the contextualized queries collected from movie scripts (*i.e.*, LSMDC), than the late dot-product fusion in [5, 11]. Furthermore, competitive contrastive methods [5, 11, 19, 43] may offer fast inference for retrieval, but cannot support QA and captioning with the same model weight. How to find a balance between fast retrieval and unification is an interesting future direction.[5]

5. Conclusion

We introduce LAVENDER, the first unified video-language (VidL) framework, that can tackle various VidL tasks with a unified Masked Language Modeling objective. Without any task-specific architectures, LAVENDER outperforms the prior state-of-the-art on 11 out of 14 benchmarks considered. Experiments show that LAVENDER is better suited for multi-task learning, few-shot generalization and zero-shot evaluation on video question answering tasks. Promising future extensions of LAVENDER include: (*i*) extension to fine-grained VidL tasks (*e.g.*, video corpus moment retrieval [31]); and (*ii*) more effective in-context few-shot learning or prompt tuning. Like other data-driven systems, LAVENDER shares similar risks that may have negative societal impact, such as biases in training data and energy consumption with large-scale training. However, we believe that our unified framework combined with multi-task learning can most likely reduce both memory and energy costs, and potentially lead to more economical deployment in real-world applications.

[5]We also include a brief comparison of ZS retrieval on DiDeMo, and show that LAVENDER is also competitive (ours, R1/R5: 22.4/47.3 vs. VideoCLIP [70], R1/R5: 16.6/46.9).

References

[1] Hassan Akbari, Linagzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. VATT: Transformers for Multimodal Self-Supervised Learning from Raw Video, Audio and Text. In *NeurIPS*, 2021. 2

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 3, 7

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval. In *ICCV*, 2021. 2, 4, 8

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 3

[5] Simion-Vlad Bogolin, Ioana Croitoru, Hailin Jin, Yang Liu, and Samuel Albanie. Cross modal retrieval with querybank normalisation. *arXiv preprint arXiv:2112.12777*, 2021. 2, 8

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 7

[7] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In *CVPR*, 2021. 4

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

[9] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO Captions: Data Collection and Evaluation Server. In *arXiv preprint arXiv:1504.00325*, 2015. 2, 3, 4

[10] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV*, 2020. 1, 2

[11] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. In *AAAI*, 2021. 8

[12] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *ICML*, 2021. 2, 3, 6

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019. 1, 3, 5

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. 3

[15] 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. *arXiv preprint arXiv:2111.12681*, 2021. 2, 3, 4, 5, 8

[16] Tsu-Jui Fu*, Linjie Li*, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. An empirical study of end-to-end video-language transformers with masked visual modeling. In *CVPR*, 2023. 2

[17] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal Transformer for Video Retrieval. In *ECCV*, 2020. 2

[18] Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, and Jianfeng Gao. Vision-language pre-training: Basics, recent advances, and future trends. *arXiv preprint arXiv:2210.09263*, 2022. 2

[19] Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo. Bridgeformer: Bridging video-text retrieval with multiple choice questions. In *CVPR*, 2022. 8

[20] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, 2017. 3, 7

[21] Tanmay Gupta, Amita Kamath, Aniruddha Kembhavi, and Derek Hoiem. Towards general purpose vision systems. In *CVPR*, 2022. 2

[22] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing Moments in Video with Natural Language. In *ICCV*, 2017. 1, 2, 5

[23] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *ICCV*, 2021. 2

[24] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. In *CVPR*, 2017. 1, 2, 5

[25] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, and Tom Duerig Zhen Li. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *ICML*, 2021. 1

[26] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset. In *arXiv preprint arXiv:1705.06950*, 2017. 3, 5

[27] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 2, 4

[28] Jie Lei, Tamara L Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. *arXiv preprint arXiv:2206.03428*, 2022. 2

[29] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is More: ClipBERT for Video-and-Language Learning via Sparse Sampling. In *CVPR*, 2021. 1, 2, 3, 5, 8

[30] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. TVQA: Localized, Compositional Video Question Answering. In *EMNLP*, 2018. 2

[31] Jie Lei, Licheng Yu, Tamara L. Berg, and Mohit Bansal. TVR: A Large-Scale Dataset for Video-Subtitle Moment Retrieval. In *ECCV*, 2020. 2, 8

[32] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 7

[33] 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. 1, 2, 4

[34] Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, Yu Cheng, Luowei Zhou, Xin Eric Wang, William Yang Wang, Tamara Lee Berg, Mohit Bansal, Jingjing Liu, Lijuan Wang, and Zicheng Liu. VALUE: A Multi-Task Benchmark for Video-and-Language Understanding Evaluation. In *NeurIPS*, 2021. 2, 6

[35] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *ECCV*, 2020. 1, 2

[36] 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 *CVPR*, 2022. 2, 3, 4, 8

[37] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In *arXiv preprint arXiv:1907.11692*, 2019. 1, 3

[38] 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 *ICCV*, 2021. 3

[39] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video Swin Transformer. In *CVPR*, 2022. 3, 5

[40] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*, 2019. 1

[41] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-Task Vision and Language Representation Learning. In *CVPR*, 2020. 6

[42] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. 2

[43] 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. In *arXiv preprint arXiv:2104.08860*, 2021. 8

[44] Tegan Maharaj, Nicolas Ballas, Anna Rohrbach, Aaron Courville, and Christopher Pal. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *CVPR*, 2017. 2, 5

[45] 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. 2

[46] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019. 2

[47] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*, 2011. 4

[48] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, Joao Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *ICLR*, 2021. 2

[49] 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 *ICML*, 2021. 1, 8

[50] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. In *arXiv preprint arXiv:1910.10683*, 2020. 1

[51] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A Dataset for Movie Description. In *CVPR*, 2015. 1, 2, 5

[52] Andrew Rouditchenko, Angie Boggust, David Harwath, Brian Chen, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Hilde Kuehne, Rameswar Panda, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, and James Glass. AVLnet: Learning Audio-Visual Language Representations from Instructional Videos. In *INTERSPEECH*, 2021. 2

[53] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021. 6

[54] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. In *CVPR*, 2022. 2, 8

[55] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *ACL*, 2018. 2, 4

[56] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *CVPR*, 2022. 2

[57] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *ICLR*, 2020. 2

[58] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A Joint Model for Video and Language Representation Learning. In *ICCV*, 2019. 1, 2

[59] Hao Tan and Mohit Bansal. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *EMNLP*, 2019. 1, 2

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

[61] 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. 3

[62] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. In *CVPR*, 2023. 7, 8

[63] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv:2202.03052*, 2022. 2, 3

[64] 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. 1

[65] Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, et al. Language models with image descriptors are strong few-shot video-language learners. In *NeurIPS*, 2022. 7

[66] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *ICLR*, 2022. 2, 3

[67] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. In *arXiv preprint arXiv:1609.08144*, 2016. 3

[68] 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 *ACMMM*, 2017. 1, 2, 5

[69] Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Masoumeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. Vlm: Task-agnostic video-language model pre-training for video understanding. In *Findings of ACL-IJCNLP*, 2021. 2

[70] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. VideoCLIP: Contrastive Pre-training for Zero-shot Video-Text Understanding. In *EMNLP*, 2021. 2, 8

[71] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*, 2016. 1, 2, 5

[72] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *CVPR*, 2022. 3

[73] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just Ask: Learning to Answer Questions from Millions of Narrated Videos. In *ICCV*, 2021. 2, 7, 8

[74] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. In *NeurIPS*, 2022. 3, 7

[75] Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. TACo: Token-aware Cascade Contrastive Learning for Video-Text Alignmentl. In *ICCV*, 2021. 2

[76] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *ECCV*, 2022. 2, 3

[77] Weijiang Yu, Haoteng Zheng, Mengfei Li, Lei Ji, Lijun Wu, Nong Xiao, and Nan Duan. Learning from inside: Self-driven siamese sampling and reasoning for video question answering. In *NeurIPS*, 2021. 2

[78] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, 2018. 5

[79] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *CVPR*, 2022. 2, 7

[80] 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 *NeurIPS*, 2021. 1, 2, 3, 5, 8

[81] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *CVPR*, 2021. 2

[82] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards Automatic Learning of Procedures from Web Instructional Videos. In *AAAI*, 2018. 2

[83] Linchao Zhu and Yi Yang. ActBERT: Learning Global-Local Video-Text Representations. In *CVPR*, 2020. 1, 2