Supplementary Material for 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

A. Additional Results

Results on Image-text Tasks. As the design of LAVENDER is not specific to video-text inputs, we can extend it to image-text tasks. Table 1 summarizes some initial results by evaluating the pre-trained LAVENDER on a challenging image-text task Visual Commonsense Reasoning [36] (VCR), following [9, 37]. LAVENDER can still perform competitively on this image-text task, much better than MERLOT pre-trained on 3M images or 100M videos (71.6 vs. 58.9/66.3). Note that LAVENDER is pre-trained and finetuned with lower-resolution input (224x224), while MERLOT is based on videos/images of much higher resolution (384x704).

Method	# pre-train videos/images	# pre-train epochs	VCR Q→A
MERLOT [37]	-/∼3M	5	58.9
	100M/-	5	66.3
	180M/-	5	75.2
	180M/-	40	80.6
VIOLET [9]	183M/3M	5	76.3
LAVENDER	2.5M/3M	10	71.6

Table 1. **Evaluation on Visual Commonsense Reasoning** [36]. Top-1 accuracy is reported for the sub-task Q→A.

Results with Other Video Backbone. For strictly fair comparison, we design LAVENDER to be only different from LAVENDER-TS on the shared Masked Language Modeling (MLM) head and objective. Hence, results in Table 2 of the main text provide evidence that the performance gain largely comes from the unified architecture. Here, we conduct additional experiments with shared MLM head and objective based on the ClipBERT [15] architecture, where a ResNet-50 [11] with mean pooling is used for video encoder, text encoder + fusion encoder is similarly initialized with BERT-base. We follow [15] to pre-train on COCO [6]+VG [14] data with our proposed VTM as MLM + MLM, and perform single-task finetuning on downstream tasks. Results in Table 2 further validates the gain from our unified architecture.

Method	Pre-train Task	TGIF Act.	MSRVTT QA	DiDeMo Ret
ClipBERT [9]	VTM+MLM	82.9	37.4	43.1
Ours	VTM as MLM + MLM	**88.9**	**40.2**	**43.8**

Table 2. **Results with other video backbone**. We experiment with ResNet-50 + Mean pooling for the video encoder, as proposed in ClipBERT [15]. Both models are pre-trained on COCO [6]+VG [14].

Results with Frozen Multimodal Encoder. We follow the standard practice and the popular trends in the literature [9, 15, 37] to train LAVENDER in an end-to-end manner, during both pre-training and finetuning stages. In Table 3, we compare model performance of frozen multimodal encoder and end-to-end finetuning with both LAVENDER-TS and LAVENDER. Freezing encoder parameters results in severe performance drop for both models (-32.8 for LAVENDER and -34.0 for LAVENDER-TS).

Method	Freeze Encoder	Meta Ave.	TGIF Act.	MSVD QA	MSRVTT Cap.	DiDeMo Ret
LAVENDER	Y	36.1	28.1	37.9	33.4	44.8
	N	**68.9**	**95.8**	**54.4**	**57.3**	**68.2**
LAVENDER-TS	Y	30.0	21.7	19.2	34.8	44.3
	N	**64.0**	**94.5**	**46.7**	**59.0**	**55.7**

Table 3. **Results with frozen multimodal encoder**. All results are reported with single-task finetuning, based on the pre-trained weights on 2.5M videos + 3M images.

Ablation on the position to insert `[MASK]`. We ablate the position to insert `[MASK]` token during both pre-training and finetuning, with 4 variants:

- **Replace** `[CLS]` with `[MASK]`.
- Insert `[MASK]` at the **begin**ning of the sentence before `[CLS]`.
- Insert `[MASK]` in the **middle** of the sentence. For simplicity, we insert the `[MASK]` token at fixed position as the 10th token.

Finetune	Method	Meta	TGIF			MSRVTT				LSMDC			MSVD			DiDeMo
		Ave.	Act.	Trans.	Frame	MC	QA	Ret	Cap	MC	FiB	Ret	QA	Ret	Cap	Ret
MT (all-in-one)	LAVENDER-TS	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
	LAVENDER	**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)	LAVENDER-TS	69.6	93.8	97.4	65.9	92.2	41.7	52.7	54.2	83.1	49.8	34.8	50.9	65.6	135.8	56.5
	LAVENDER	**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**

Table 4. **Comparison between LAVENDER and LAVENDER-TS under full multi-task setting**. 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. All results are reported under VidL pre-training on 2.5M videos + 3M images.

Finetune		Meta	TGIF			MSRVTT				LSMDC			MSVD			DiDeMo
Method	# Params	Ave.	Act.	Trans.	Frame	MC	QA	Ret	Cap	MC	FiB	Ret	QA	Ret	Cap	Ret
ST	14P	76.0	94.8	98.7	73.5	97.2	**45.0**	61.4	59.4	85.9	**57.1**	41.9	55.6	**72.3**	150.3	**72.4**
MT (all-in-one)	P	75.2	95.4	98.4	71.0	94.5	44.7	58.9	57.9	**87.0**	56.6	41.7	54.0	71.7	150.0	71.1
MT (best)	14P	75.4	95.6	98.6	72.5	94.5	44.8	59.1	58.9	**87.0**	56.9	41.9	**56.6**	71.9	150.1	71.3
MT → ST	14P	76.2	**96.3**	98.6	71.8	**97.4**	**45.0**	**61.7**	**60.1**	**87.0**	**57.1**	**43.3**	54.3	71.8	**150.7**	71.5

Table 5. **Multi-task Finetuning of LAVENDER with scale-up VidL Pre-training on 14M videos + 16M images**. 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 (backbone + MLM head).

- Insert `[MASK]` at the **end** of the sentence. This is the default setting in the paper.

	Replace	Begin	Middle	End
Replace	50.4	50.2	50.0	50.6
Begin	51.1	51.2	50.7	51.8
Middle	48.7	48.6	50.3	50.6
End	50.9	51.2	51.7	**52.2**

(a) MSVD-QA

	End
Replace	54.6
Begin	54.6
Middle	53.8
End	**55.5**

(b) MSRVTT-Cap

	Replace	Begin	Middle	End
Replace	71.6	78.7	91.9	91.7
Begin	83.0	80.8	92.9	92.4
Middle	91.5	90.7	**93.3**	91.3
End	90.7	91.6	91.8	92.9

(c) TGIF-Action

Table 6. **Ablations on the position to insert `[MASK]`** during both pre-training (row of each table) and finetuning stage (column of each table). All results are based on pre-trained weights on 3M images with single-task fientuning. Note that for auto-regressive caption generation on MSRVTT-Cap, the `[MASK]` token is always appended to previously generated tokens.

For faster iteration, we pre-train LAVENDER by varying the `[MASK]` position on CC3M data, and perform single-task finetuning on each task. Results in Table 6 show that inserting `[MASK]` at the end of the sentence brings competitive performance consistently over different tasks.

Ablation on Pre-training Data. We conduct ablations on using image-text only (CC3M [26]) or video-text only (WebVid2.5M [2]) data for pre-training, and compare it with using both datasets for pre-training in Table 7. Compared to without pre-training, image-text pairs alone pre-training improves on three tasks and performs comparably on TGIF-Action. Video-text pairs alone pre-training improves on all tasks, and combining image-text and video-text together achieves the best results. Our observation of this combined pre-training recipe being beneficial for video-text tasks is consistent with what was reported in [2].

Pre-train Data	Meta Ave.	TGIF Act.	MSVD QA	MSRVTT Cap.	DiDeMo Ret
N/A	45.5	93.5	40.8	47.7	0.0
WebVid2.5M	65.1	94.3	53.0	54.7	58.2
CC3M	65.4	92.9	52.2	55.5	61.1
WebVid2.5M+CC3M	**68.9**	**95.8**	**54.4**	**57.3**	**68.2**

Table 7. **Ablation on pre-training data**. All results are reported with single-task finetuning.

Model Architecture	# Layers in Decoder	MSRVTT-Cap
Encoder-only (LAVENDER)	N/A	**47.7**
Encoder-decoder	2	42.6
	4	45.0
	6	43.8
	12	42.8

Table 8. **Comparison between encoder-only and encoder-decoder architecture**. All results are based on single-task finetuning without pre-training.

Encoder-only vs. encoder-decoder architecture. We follow the popular model architecture adopted in video-language literature, which is encoder-only architecture [9, 15, 27, 34, 37]. Compared to encoder-decoder architecture, MLM head is more lightweight, as shown in Figure 1 of the main text. That being said, we report performance of an encoder-decoder model on MSRVTT captioning as a comparison in Table 8. The results show some interesting findings: (i) reducing the number of decoder layers can improve caption performance (CIDEr score), but the improvements become less prominent when using only 2 de-

coder layers; (ii) encoder-only achieves better performance than the encoder-Decoder variants, which may due to more randomly initialized parameters added in Encoder-Decoder architecture. Full encoder-decoder model pre-training is out of the scope of this paper.

MSRVTT	QA	Caption	Retrieval
ST	**44.2**	57.3	**58.9**
MT (video domain)	44.1	56.8	55.3
MT (task type)	43.2	56.9	-
MT (mixed, as in Table 2, main text)	-	**57.4**	-
All-in-one (as in Table 3, main text)	**44.2**	57.2	56.4

Table 9. Investigations on **different multi-task settings** with LAVENDER. Due to differences in data split for MSRVTT tasks, we strictly filter out testing videos from all training splits for MT (Appendix C). Hence, on retrieval task, ST model is finetuned with more data than MT/All-in-one models.

Investigation on Other Multi-task Settings. We follow previous work [17] to explore different MT settings and share similar findings. We compare MT by video domain/task type with mixing domain and task type (Table 2, main text) and All-in-one (Table 3, main text). Overall, All-in-one empirically strikes a balance between sophisticated heuristic designs of multi-task setting and good model performance.

Full Results on Multi-task Finetuning. Table 4 compares LAVENDER and LAVENDER-TS under full multi-task finetuning settings, LAVENDER consistently outperforms task-specific baseline over all tasks with a gain of +4.2 on average. For completeness, we include additional results under multi-task finetuning. Table 5 presents the results of single-task finetuning and multi-task variants from the scale-up pre-training on 14M videos + 16M images. For easier reference in future work, we report detailed retrieval results on R1/5/10 in Table 10.

Investigation on Other Pre-training Tasks. As mentioned in the main text, we only adopt Masked Language Modeling (MLM) and Video Text Matching (VTM) as pre-training tasks for both the proposed LAVENDER and the task-specific baseline LAVENDER-TS. Here we briefly discuss other popular pre-training objectives with LAVENDER-TS. The first is **Frame Order Modeling** [16,37], where the input video frames are randomly shuffled and the goal is to revert back its original order. Different from the video-ASR pairs utilized in these works, the paired text in our pre-training data is not temporally grounded. In most cases, the shuffled frame sequence will probably still be globally aligned with the textual description. Hence, such fine-grained temporal reasoning objective is not applicable in our case. The second is **Masked Visual Modeling** (MVM), where the model learns to reconstruct high-level semantics or low-level details for a certain percentage of "masked" visual inputs (*i.e.*, features or patches). Different variants have been proposed and shown little-to-none effect in vision-language pre-training, such as predicting the object category of masked image regions [7] and distilling region/frame features from well-supervised vision encoders [7, 16]. More recently, by taking advantage of pre-trained DALL-E [23], researchers [3, 9, 28] have shown potentials in masked visual token modeling, which asks the model to recover the discrete latent codes of the masked image patches. [31] explores image feature descriptors such as Histograms of Oriented Gradients (HOG) as the prediction target for self-supervised visual pre-training. In Table 11, we investigate three different MVM objectives on top of VTM + MLM pre-training for LAVENDER-TS: (i) VQ Token: to recover the discrete codes extracted from pre-trained DALL-E following [9]; (ii) Pixel: to regress the RGB colors as in [31]; and (iii) HOG: to regress the HOG values, following [31]. Results show that only MVM with HOG achieves a marginal performance improvement of +0.3 on average. Therefore, we adopt a simple recipe for all other pre-training experiments in the paper, that is with only MLM and VTM.

Qualitative Comparisons to Task-specific Baseline. Figure 1 provides qualitative comparisons between LAVENDER and task-specific baseline LAVENDER-TS on video question answering (QA). The model predictions are sampled from MSVD-QA.

In Figure 1a, the ground-truth answer "fold" is not in the top-k ($k = 1000$ for MSVD-QA, following [9]) most common answers in training split, hence excluded from the pre-defined answer vocabulary for training LAVENDER-TS. In Figure 1b, the ground-truth answer "bowl" appears roughly 9 times more than "bag" in the training split. These visualization results on video QA suggest that (i) our LAVENDER can better fit the open-ended setting for QA tasks, as it does not restrict the predictions to be from a pre-defined answer vocabulary as in LAVENDER-TS (Figure 1a); and (ii) the task-specific baseline is easier to fail on questions with out-of-distribution answers than LAVENDER (Figure 1b). Additionally, we show in Figure 1c when both models can provide reasonable answers to the question, which do not exactly match the ground-truth answer. This result reveals potential problems with the current evaluation metrics or existing datasets on video QA. Future work may consider collecting additional annotations to enrich the dataset and improve the evaluation metric to handle multiple ground-truth answers (*e.g.*, similar to VQA scores [10]). Other common failure cases of LAVENDER may result from sparsely sampled visual inputs. When the key frames are missing, the model fails on QA or retrieves a wrong video or generates inaccurate captions, which unarguably is a shared caveat among existing SOTA VidL models.

B. Additional Comparison with Existing Work

Figure 2 in the main text shows the detailed comparison between LAVENDER and existing methods with the

# Pretrain videos/images	Finetune Method	Text-to-Video Retrieval			
		MSRVTT	DiDeMo	MSVD	LSMDC
2.5M / 3M	ST	**37.8** / **63.8** / 75.0	**47.4** / **74.7** / 82.4	45.8 / 75.7 / 85.0	22.2 / **43.8** / **53.5**
	MT (all-in-one)	33.7 / 61.3 / 73.9	44.1 / 72.4 / 81.8	45.4 / 76.5 / 85.8	22.4 / 42.8 / 53.1
	MT (best)	34.9 / 61.1 / 73.2	46.3 / 72.2 / 81.1	46.2 / 76.2 / 85.5	22.4 / 42.8 / 53.1
	MT → ST	36.8 / 63.4 / **75.2**	45.9 / 73.0 / **84.1**	**46.3** / **76.9** / **86.0**	**22.8** / 43.5 / 53.2
14M / 16M	ST	39.7 / 66.7 / **77.8**	**53.4** / **78.6** / **85.3**	**50.1** / **79.6** / **87.2**	25.1 / 44.6 / 56.0
	MT (all-in-one)	37.6 / 63.4 / 75.6	50.3 / 77.8 / 85.2	49.4 / 78.7 / 86.8	23.4 / 77.8 / 85.2
	MT (best)	37.5 / 64.2 / 75.7	51.0 / 78.1 / 84.7	49.9 / 79.0 / 86.8	24.1 / 46.3 / 55.4
	MT → ST	**40.7** / **66.9** / 77.6	51.6 / 77.7 / 85.1	49.8 / 78.8 / 86.8	**26.1** / **46.4** / **57.3**

Table 10. **Detailed results of LAVENDER on text-to-video retrieval tasks** under single-task (ST) finetuning, and different multi-task (MT) finetuning settings. All results are reported on R1/5/10.

Pre-raining Tasks	Meta Ave.	TGIF			MSRVTT			LSMDC	MSVD	DiDeMo
		Act.	Trans.	Frame	QA	Ret	Cap	FiB	QA	Ret
VTM+MLM	60.9	91.5	**98.6**	**64.6**	**40.7**	50.6	53.0	**51.9**	45.3	52.2
+ VQ Token [9]	60.8	**92.4**	**98.6**	63.9	40.3	**52.1**	52.5	51.1	44.5	51.6
+ Pixel [31]	60.3	91.0	98.4	63.4	40.5	52.3	50.7	51.6	42.5	52.2
+ HOG [31]	**61.2**	91.7	**98.6**	64.4	40.4	51.8	**53.4**	50.5	**45.8**	**53.9**

Table 11. We investigate different **Masked Visual Modeling** tasks for pre-training LAVENDER-TS. 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. VidL pre-training is conducted on WebVid2.5M [2].

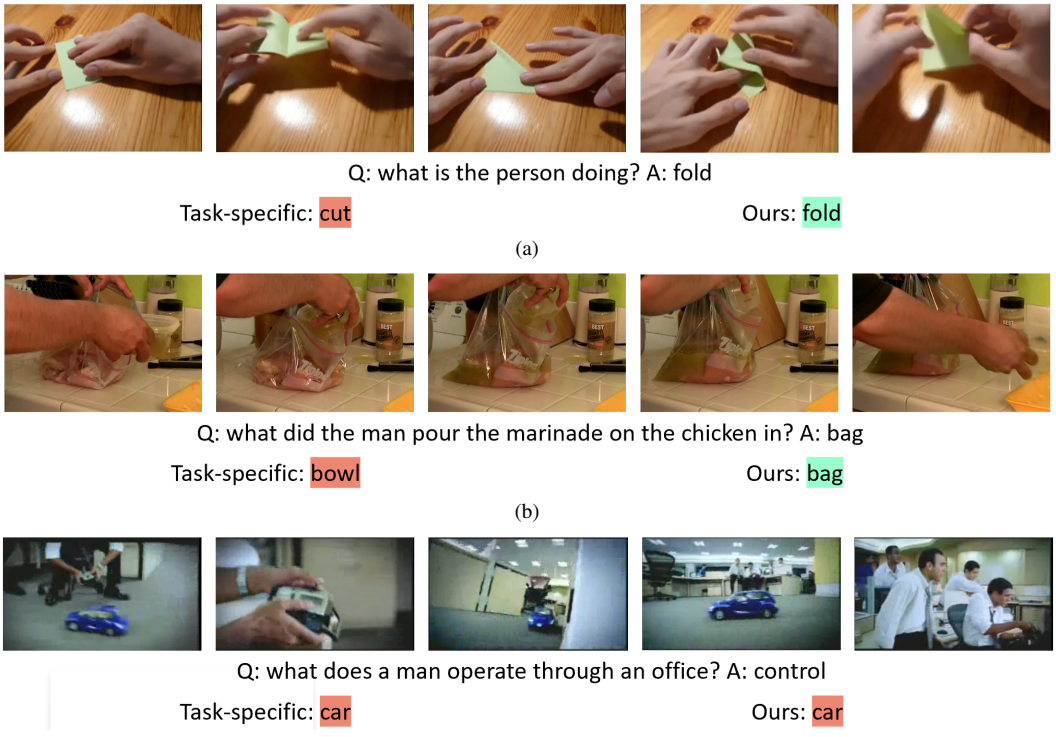
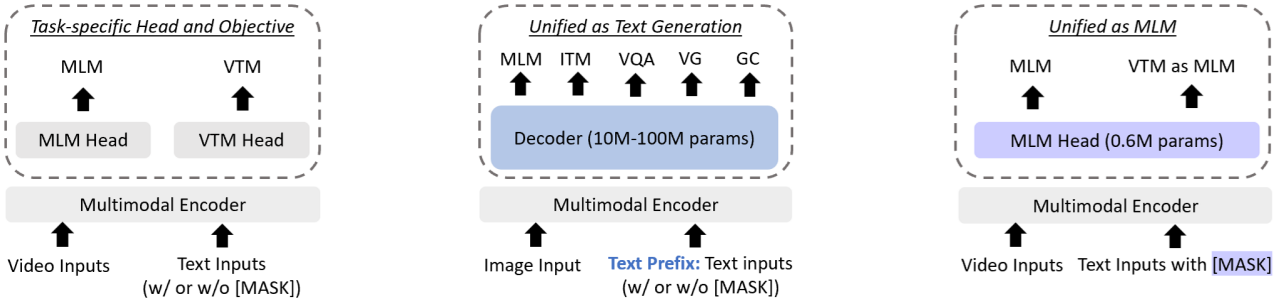


Figure 1. **Visualization of model predictions from LAVENDER (ours) and LAVENDER-TS (task-specific)** on video question answering. Green (Red) highlights the correct (wrong) predictions.

image/video question answering task as an example. In Figure 2, we illustrate the differences among these methods in pre-training. Unlike existing video-language models, which design task-specific heads and objectives for different pre-training tasks. LAVENDER unifies masked language modeling (MLM) and video text matching (VTM) as MLM. Compared with unified image-text models (*e.g.*, VL-T5 [8]), which are typically pre-trained with a combination of complex pre-training tasks, such as visual question answering and grounded captioning. Although these pre-training tasks



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

Figure 2. Illustration of the **differences between LAVENDER and existing methods** during pre-training. LAVENDER unifies both masked language modeling (MLM) and video text matching (VTM) as MLM, without task-specific heads in existing video-language (VidL) models. Take VL-T5 [8] as an example; most unified image-text models are pre-trained with a combination of complex pre-training tasks (*e.g.*, visual question answering (VQA), visual grounding (VG), grounded captioning (GC)).

may enable the model with new abilities (*e.g.*, generating region proposals as in [30, 35]), the supervision often comes from human-annotated data. It remains unclear how to design and effectively pre-train a unified model with such capability but without dependency on human-labeling.

C. Implementation Details

Task-specific Prompts and Tokens. As mentioned in Section 4.3 of the main text, we explore the vanilla multi-task finetuning without any task-specific designs and two additional variants with task-specific prompts and tokens for LAVENDER. Here, we describe the prompts and tokens used in these baselines.

For **task-specific prompts**, we insert a human-readable text prompt at the beginning of the text input.

- For text-to-video retrieval and video-text matching during pre-training, the text prompt is *"is the video-text paired, true or false"*;
- For multiple-choice video question answering (QA), the text prompt is *"which answer choice is correct, choose from 0, 1, 2, 3, 4."*;
- For open-ended video QA, the text prompt is *"answer the question about the video."*;
- For video captioning, the text prompt is *"write a description about the video."*.

As discussed in the Experiments section of the main text, we only briefly investigate prompt tuning with LAVENDER. How to design more diverse prompts for more effective prompt tuning is an interesting direction for future work.

For **task-specific tokens**, we add several new tokens to the whole vocabulary, and learn these token embeddings from scratch during multi-task finetuning. For both training and inference, the task-specific token is inserted right after `[CLS]` token in the text input. Specifically, we add `[VTM]` for text-to-video retrieval and video-text matching, `[MC]` for multi-choice video QA, `[OE]` for open-ended video QA, `[CAP]` for video captioning.

Additional Training Details. All experiments are conducted on Microsoft Azure [1], adopting mixed-precision training with DeepSpeed [24]. All video data are pre-processed by evenly extracting 32 frames to avoid expensive decoding on-the-fly. Our implementation of LAVENDER is based on PyTorch [22]. We adopt AdamW [18] as the optimizer with an initial learning rate of 2e-5, betas of (0.9, 0.98), and weight decay of 1e-3 for all pre-training and finetuning experiments. For pre-training, we adopt a batch size of 28 per GPU. During training, we randomly sample T frames from 32 frames, resize the shorter side of all frames to 224 and random crop (H=W=224) at the same location for all the frames in a given video, to further split into patches with size h=w=32. During inference, we evenly sample T frames from 32 frames and center crop (H=W=224) for all the frames. For all downstream tasks, we adopt the same video frame size and patch size, but T=5 video frames. We summarize the training configurations for downstream finetuning in Table 12. Due to various data scales and domains, we use task-specific batch size and training epochs based on the performance of the validation set for each downstream task (Table 12b). All other settings are shared across all datasets (Table 12a).

For multi-task finetuning, since the same set of videos are shared among several downstream tasks, there might be overlaps between one's training split and others' validation or testing split (*e.g.*, some video-text pairs in MSRVTT-Retrieval 9K-train is in the testing split of MSRVTT-Captioning). To avoid data contamination, we filter out validation and testing videos in all downstream datasets from the training splits, and use this cleaned version for multi-task finetuning. At each training step, we randomly sample one dataset from all 14 of them, and construct a batch of examples from that dataset. The training is conducted on

16×80GB A100 for 20 epochs, and we adopt the same batch size for retrieval tasks as shown in Table 12b, and batch size 60 for all other tasks.

Learning Rate	2e-5
Weight Decay	1e-3
Optimizer	AdamW [18]
βs	(0.9, 0.98)
Warmup Ratio	10%
# Frames (T)	5
Frame Size (H, W)	(224, 224)
Patch Size (h, w)	(32, 32)

(a) Common Configurations.

Dataset	# GPUs	Batch Size / GPU	# Epochs
Video Question Answering			
TGIF-Action	8×32GB V100	24	56
TGIF-Transition			15
TGIF-Frame			10
MSRVTT-QA			8
LSMDC-MC			10
LSMDC-FiB			5
MSVD-QA			8
Text-to-Video Retrieval			
MSRVTT	16×80GB A100	20	10
LSMDC	8×80GB A100		5
MSVD		16	5
DiDeMo			10
Video Captioning			
MSRVTT	8×32GB V100	24	20
MSVD			

(b) Task-specific Configurations.

Table 12. Training Configurations For Downstream Finetuning.

D. Pre-training Data

Public Datasets We use the following publically available datasets to pretrain LAVENDER:

- **WebVid2.5M** [2] scrapes 2.5M video-text pairs from the web. The texts in this data are alt-text descriptions, which generally describe the global video semantics.
- **Conceptual Captions 3M** (CC3M) [26] consists of 3.3M image-text pairs, which are also harvested from the web. **CC12M** [4] further enlarges CC3M by 4 times. Both have been used to pre-train large-scale image-text models.
- **SBU-Captions** [21] is another widely used dataset for image-text pre-training, web-crawled from Flickr. It contains 1M image-text pairs.
- **COCO** [6] and **Visual Genome** (VG) are two human-annotated image-text datasets. COCO contains 5 captions per image over 120K images. Unlike COCO captions that can describe the whole scene, VG collects 5M regional descriptions over more than 100K images.

Video-Text Data Collection For the scale-up pre-training, we additionally crawl 11.9M video-text pairs from the web, following the same procedure in [2]. Here, we briefly describe how we collected the data.

WebVid2.5M has led to promising results in text-to-video retrieval tasks as shown in [2]. This motivates us to further crawl more video-text pairs from the same source. We first use a search engine to identify the potential data sources based on sampled textual descriptions in WebVid2.5M, and then we scrape the video-text pairs from these data sources. Similarly, we follow [2, 26] to filter out offensive content and hide person and location names. In total, we have collected 11.9M videos, each accompanied with an alt-text description. The collected dataset shares similar characteristics as WebVid2.5M, with the average video duration as $\sim$ 20 seconds, and the average number of words in the textual description as $\sim$20. Note that at the time when we started this project, WebVid10M in [2] has not been released yet. We later found our 11.9M data largely overlaps with WebVid10M. Hence, we refer future work to WebVid10M for scale-up pre-training.

E. Downstream Datasets

In this section, we introduce all downstream datasets used for evaluating LAVENDER and discuss some dataset-specific training details below. Table 13 summarizes the number of examples in training/validation/testing split for each dataset.

Text-to-video Retrieval We evaluate LAVENDER on 4 popular text-to-video retrieval datasets, namely MSRVTT [33], DiDeMo [12], MSVD [5] and LSMDC [25]. **MSRVTT** contains 10K YouTube videos with 200K descriptions. We follow [2] to train on 9K videos and evaluate on 1K-A testing split. **DiDeMo** consists of 10K Flickr videos, each annotated with 4 sentences. We concatenate all sentences from the same video into a paragraph and perform paragraph-to-video retrieval, following [2, 15]. Although this dataset comes with localisation annotations (ground-truth temporal proposals) for each sentence, we perform all experiments without leveraging this fine-grained information for both training and evaluation. Instead, we use the same procedure as described in Appendix C to sample frames from videos. **MSVD** is based on 2K YouTube videos and crowdsourced 40 textual descriptions per video. **LSMDC** is built upon 118K video clips from 202 movies. Each clip has a caption from movie scripts or descriptive video services. We use the standard splits for DiDeMo, MSVD and LSMDC, following [19]. For paragraph-to-video retrieval on DiDeMo, we adopt the text augmentation technique proposed in Frozen [2], which is to randomly sample and concatenate a variable number of sentences as paragraph for each video.

Multiple-choice Video QA We evaluate LAVENDER on four multiple-choice QA datasets: TGIF-Action, TGIF-Transition [13], MSRVTT-MC [32] and LSMDC-MC [20].

	TGIF			MSRVTT		LSMDC		MSVD
	Action	Transition	Frame	QA	MC	FiB	MC	QA
Training	18K / 18K	26K / 47K	30K / 35K	6.5K / 149K	- / -	95K / 297K	101K / 101K	1.2K / 30K
Validation	2K / 2K	5K / 5K	4K/ 4K	0.5K / 123K	- / -	7K / 22K	7K / 7K	0.2K / 6K
Testing	2K / 2K	3K / 6K	7K/ 14K	3K / 73K	3K / 3K	9.5K / 30K	10K/ 10K	0.5K / 13K
# answer choices (MC-QA)	5	5	-	-	5	-	5	-

(a) Video Question Answering Tasks (# videos / # video-question pairs). For open-ended QA, we do not restrict the answer vocabulary to contain only the most common answers in training split. Theoretically, the model predictions can be any word in the whole vocabulary of `vocab_size` = 30,522.

	MSRVTT		MSVD		LSMDC	DiDeMo
	Ret.	Cap.	Ret.	Cap.	Ret	Ret
Training	9K / 180K	6.5K / 130K	1.2K / 49K	1.2K / 49K	101K / 101K	8K / 8K
Validation	1K / 1K†	0.5K / 10K	0.1K / 4K	0.1K / 4K	7K / 7K	1K / 1K
Testing	1K / 1K	3K / 60K	0.7K / 28K	0.7K / 28K	1K / 1K	1K / 1K

(b) Text-to-video Retrieval and Video Captioning Tasks (# videos / # video-text pairs).†: on MSRVTT-Retrieval, we use the same split of 9K-training in [2, 19]. For the validation purpose, we use the original validation split in 7K-training version, whose examples are included in the training split of 9K-training.

Table 13. **Data Distribution of Downstream Datasets**.

	TGIF-Frame			MSRVTT-QA			LSMDC-FiB			MSVD-QA		
	Train	Val	Test	Train	Val	Test	Train	Val	Test	Train	Val	Test
Max answer_len	4	5	5	4	7	6	3	3	3	4	4	4
% of data w/ answer_len > 1	2.8	2.8	2.5	2.9	4.9	4.6	2.4	2.7	2.6	6.6	7.9	7.0

Table 14. **Answer Length Distribution for Open-ended Video Question Answering**. In summary, there are $< 8\%$ of examples across training, validation and testing split of each dataset, with answer length > 1.

Among them, **TGIF-Action** and **TGIF-Transition** aim to test the model's ability to recognize repeating actions and state transitions in short GIFs. Each video-question pair is accompanied with 5 answer choices. We concatenate the 5 answer choices sequentially with the question, and the model is asked to predict the ground-truth answer index. **MSRVTT-MC** and **LSMDC-MC** are based on retrieval tasks, but reformulated as multiple-choice QA. A model needs to find the caption that describes the video out of 5 candidate captions. Due to its similarity to video-to-text retrieval, we formulate it as video-text matching, which is the same as zero-shot evaluation described in the Experiments section of the main text. 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. As there is no training and validation data constructed in the same way for MSRVTT-MC, we follow [15] to evaluate the retrieval model trained on MSRVTT to rank the 5 candidate answers.

Open-ended Video QA Four datasets are considered for open-ended video QA: TGIF-Frame [13], MSRVTT-QA, MSVD-QA [32] and LSMDC-FiB [20]. Among them, the question-answer pairs in all but TGIF-Frame are based on the linguistic transformation of captions for each video. Questions in **TGIF-Frame** is collected via crowd-sourcing, which are answerable with just a single frame in the video. **MSRVTT-QA** contains 243K open-ended questions over 10K videos and **MSVD-QA** consists of 47K questions over 2K videos. The Fil-in-the-blank (FiB) task of **LSMDC-FiB** is, given a video and a sentence with a blank in it, to predict a correct word for the blank. We replace the blank with a `[MASK]` token, and naturally it becomes a Masked Language Modeling (MLM) task.

As mentioned in the main text, LAVENDER answers the open-ended questions in these datasets with only one word, as there is only one `[MASK]` token appended to the text input. Table 14 summarizes the max answer length and the percentage of examples with answers longer than one word in all four datasets. As the statistics show, $> 92\%$ of questions are answerable with a single word.

Video Captioning MSRVTT [33] and MSVD [5] are used for captioning evaluation. As introduced before, **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 in [33] and [29] for MSRVTT and MSVD, respectively.

The captions are generated auto-regressively during inference, while the training objective is still the same masked language modeling. During training, we randomly mask 15% of the tokens in the captions, and let the model predict the masked tokens. During inference, 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 caption generation until the model outputs a `[SEP]`, which is defined as the sentence ending token or

when it reaches the maximum generation step 50. Note, that the attention mask used for caption generation is a causal attention mask. That is, for a given word, it only attends to the words before it, not the ones coming after it.

References

[1] Microsoft Azure. https://azure.microsoft.com/. 5

[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 *ICCV*, 2021. 2, 4, 6, 7

[3] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT Pre-Training of Image Transformers. In *ICLR*, 2022. 3

[4] 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. 6

[5] David L. Chen and William B. Dolan. Collecting Highly Parallel Data for Paraphrase Evaluation. In *ACL*, 2011. 6, 7

[6] 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. 1, 6

[7] 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. 3

[8] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *ICML*, 2021. 4, 5

[9] 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. 1, 2, 3, 4

[10] 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

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 1

[12] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing Moments in Video with Natural Language. In *ICCV*, 2017. 6

[13] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. In *CVPR*, 2017. 6, 7

[14] 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. 1

[15] 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, 6, 7

[16] 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. 3

[17] 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. 3

[18] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 5, 6

[19] 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. 6, 7

[20] 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. 6, 7

[21] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*, 2011. 6

[22] 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 *NeurIPS*, 2019. 5

[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 *ICML*, 2021. 3

[24] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD*, 2020. 5

[25] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A Dataset for Movie Description. In *CVPR*, 2015. 6

[26] 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, 6

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

[28] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. VIMPAC: Video Pre-Training via Masked Token Prediction and Contrastive Learning. *arXiv:2106.11250*, 2021. 3

[29] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014. 7

[30] 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. 5

[31] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *arXiv preprint arXiv:2112.09133*, 2021. 3, 4

[32] 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. 6, 7

[33] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*, 2016. 6, 7

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

[35] 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. 5

[36] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From Recognition to Cognition: Visual Commonsense Reasoning. In *CVPR*, 2019. 1

[37] 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