Supplementary Material for VINDLU: A Recipe for Effective Video-and-Language Pretraining

Feng Cheng[1] Xizi Wang[2] Jie Lei[1] David Crandall[2] Mohit Bansal[1] Gedas Bertasius[1]
[1]UNC Chapel Hill [2]Indiana University
{fengchan,jielei,mbansal,gedas}@cs.unc.edu {xiziwang, djcran}@iu.edu

Our supplementary material consists of:

1. Additional Quantitative Results.
2. Implementation Details.
3. Dataset Descriptions.
4. Additional Temporal Modeling Baselines.

1. Additional Quantitative Results

In this section, we present the results on Action Recognition task, some useful empirical tips and additional ablation studies.

1.1. Results on Action Recognition

We finetune our pretrained video encoder on Kinetics-400 [16] directly using TimeSformer [4] codebase with exactly the same hyperparameters as in [4]. As shown in Table 1, our video encoder outperforms TimeSformer [4] and OmniVL [37] by **2.1**% and **1.0**% respectively with all models using exactly the same architecture [4]. This indicates the usefulness of our VidL pretraining recipe for a pure video understanding task.

Method	TimeSformer [4]	OmniVL [37]	VINDLU
Top-1 acc.	78.0	79.1	**80.1**

Table 1. Results on Kinetics-400 [16] for action recognition task. All models use the same TimeSformer architecture [4]. Our VINDLU approach outperforms both the TimeSformer [4] and OmniVL [37] baselines by 2.1% and 1.0% respectively. These results indicate the benefits of our VidL pretraining recipe.

Visual Encoder	MSR-VTT	DiDeMo	ANet	Avg.
ViT [3, 10]	**64.5**	**75.0**	72.9	**70.8**
VideoSwin [26]	61.1	73.1	**73.4**	69.2

Table 2. We study the performance of Isotropic (ViT) vs. Pyramid (VideoSwin) vision encoders. Based on these results, we observe that ViT outperforms VideoSwin by 1.6% on an averaged R@{1,5,10} on MSR-VTT, DiDeMo and ActivityNet-Captions. We experiment with 4 frames using our final model on the 5M corpus.

1.2. Other Useful Empirical Tips

Isotropic vs Pyramid-based Vision Encoder. Pyramid-style ViTs that use downsampling along the spatial dimension (e.g., Swin [25], MViT [11]) have shown stronger performance than isotropic ViTs (vanilla ViT) on many image/video classification tasks. Thus, several recent VidL approaches [12, 13, 24] adopt pyramid ViTs as their vision encoders. However, in our study, we find that isotropic ViTs tend to have better performance. Specifically, in Tab. 4, we show that a ViT-based encoder (ViT-B/16) outperforms VideoSwin (Swin-B) by **1.6%**. We hypothesize that this might happen because isotropic ViTs preserve more fine-grained spatial information needed for various VidL tasks.

A Linear Scaling Rule. Linear scaling strategy [14] has been extensively used for large-scale pretraining on image/video classification tasks. However, in our setting, we observed that the linear scaling rule leads to similar or worse results (See Table 3). Therefore, for all of our experiments, we use a fixed learning rate (1e-4) for all batch sizes.

Initialization. We also found that the initialization of various modules in our model is critical for good VidL performance. In particular, we note that to make MLM and MVM pretraining objectives effective, we need to use text and video encoders pretrained with these objectives in a self-supervised manner (e.g., BERT [9] and BEIT [3] respectively). Otherwise, the performance will drop significantly ($\sim$**5**% averaged R@1,5,10 accuracy drop on MSR-VTT, DiDeMo, ActivityNet datasets).

1.3. Additional Ablation Studies

MLM masking ratio. We found a larger masking ratio (50%) for the MLM objective is more helpful for VidL pretraining, compared to 15% maksing ratio used in BERT [9]. We conjecture that we can use a higher mask ratio than text-

Batch Size	512	1024	1024	2048	2048	2048
LR ($\times$1e-4)	1	1	2	1	2	4
Accuracy	68.2	68.2	68.2	**68.5**	68.3	67.4

Table 3. We investigate the effectiveness of a scaled learning rate rule [14] using averaged downstream accuracy on MSR-VTT, DiDeMo, and ActivityNet-Captions. The learning rate 1e-4 works best for various batch sizes. We experiment with 1-frame inputs using our final model on the the 5M corpus.

Masking Ratio	15%	50%	75%
Accuracy	69.2%	**70.8%**	69.9%

Table 4. We study the masking ratio for the MLM objective. We experiment with 4 frames using our final model on the 5M corpus.

	Mean Pool.	+ Temp. Attn	+ MF	+ Img Data
SSv2-{L,T}	72.3	80.2	81.3	**82.7**
M-QA	N/A	N/A	42.7	**43.6**

Table 5. The analysis for more tasks / datasets. SSv2-L and SSv2-T refers to SSv2-Label and SSv2-Template datasets [19]. M-QA refers to MSRVTT-QA [38].

only BERT because our model incorporates complementary video cues.

Analysis for More Tasks/ Datasets. We further evaluate our recipe on VidQA on MSRVTT-QA [38] and video retrieval on SSv2-Label [19], SSv2-Template [19]. As shown in Tab. 5, we report the averaged R@{1,5,10} on SSv2-* and R@1 on VidQA. Since VidQA needs a multimodal fusion (MF) encoder to generate the answers, we cannot report the results without the MF module (i.e., Columns 1,2 in Row 2 in Tab. 5). Our results indicate that our conclusions (i.e., the importance of temporal modeling, multimodal fusion, and joint image+video pre-training) also hold on these tasks/datasets.

2. Implementation Details

Positional Embeddings. We use learnable absolute temporal positional embeddings as in [2] and relative spatial positional embeddings as in [3]. The temporal positional embeddings are applied after patchifying the tokens, while the relative spatial positional embeddings are applied at each Transformer layer. When adapting the pretrained model to downstream tasks with more frames, we use zero-padding for the temporal positional embeddings as in [2]. When adapting to higher spatial resolutions, we linearly interpolate the spatial positional embeddings.

Video Retrieval. We finetune the pretrained model with

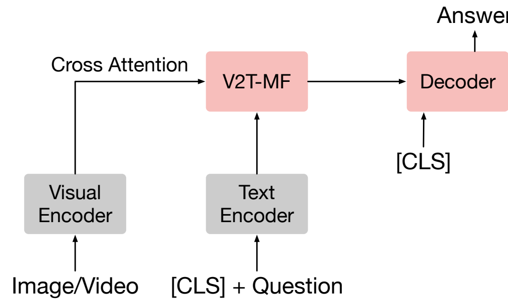

Figure 1. Our architecture for the open-ended question-answering task. The decoder uses the same architecture as our video-to-text multimodal fusion (V2T-MF) module and is initialized with the pretrained V2T-MF's weights.

VTC and VTM losses. During inference, we follow [22,23] to first select top-K ($K = 128$ in our experiments) candidates based on the video-text similarity scores of the unimodal encoders and then re-rank these candidates by calculating their pairwise VTM scores.

Open-ended Question-Answer. Following [19,23,37], we formulate this task as a text generation task. As shown in Fig. 1, we add a decoder that takes the multimodal encoder's outputs as the cross attention key and value to generate the answers. The decoder starts with a [CLS] token and ends when a [SEP] token is generated. The decoder has the same architecture as the multimodal encoder and is initialized with the pretrained multimodal encoder's weights. The model is optimized using the averaged cross-entropy loss of each token between the generated answer and the ground truth answer. For a fair comparison with prior works [19,36,37], we constrain the decoder to generate from the 3128 most common answers [19] during inference.

Multiple-Choice Question-Answering. For Multiple-Choice QA, we follow [19,23,37] and convert it to the text-to-video retrieval task. Specifically, for each question and m candidate answers, we generate m sentences by concatenating the question with each candidate's answer. We then rank these sentences by ensembling the retrieval model's video-text similarity and pairwise VTM scores. The ensembling weights are set to 0.3 for the similarity score and 0.7 for the VTM score.

Inference with More Frames. Following [19], we perform inference using more frames than our finetuned model. Specifically, we first linearly interpolate the temporal positional embeddings in the video encoder. Then all the visual tokens are concatenated and fed to the multimodal encoder.

Pretraining Datasets. As discussed in the main draft, in Steps 1-3 of our recipe, we pretrain our model on a 2M WebVid-2M [2] corpus. For Steps 4-5, we use a

Config	Pretraining	Video Retrieval				Video QA		
		MSRVTT	DiDeMo	ANet	SSv2-*	ANet	MSRVTT-QA	TVQA
optimizer	AdamW [28]							
optimizer options	$\beta_1 = 0.9, \beta_2 = 0.999$							
weight decay	0.02							
learning rate schedule	cosine decay [27]							
init learning rate	1e-4	1e-5	1e-5	1e-5	1e-4	1e-5	1e-5	1e-5
min learning rate	1e-6	1e-6	1e-6	1e-6	1e-5	1e-6	1e-6	1e-6
spatial resolution	224×224							
augementation	random resize, crop, horizontal flip							
# epochs	10	5	10	10	10	10	10	10
# warmup epochs	1	0.5	0.5	0.5	0	0	0	0
batch size $\times$ # GPUs	$64 \times \{8, 32\}$	32×4	32×1	32×1	32×2	32×1	32×1	32×1
# training frames	4	12	12	12	12	12	12	12
# inference frames	4	12	12	32	12	32	12	12

Table 6. Hyper-parameters for pretraining, and downstream tasks. SSv2-* means SSv2-Template and SSv2-Label datasets. We pretrain on 8 GPUs for C2M and C5M, 32 GPUs for C17M and C25M.

joint image-video corpus consisting of 3M images from CC3M [34] and 2M videos from WebVid-2M [2]. Lastly, in Step 6, we scale our pretraining data from $5M \rightarrow 17M \rightarrow 25M$.

Model Details. Our final VINDLU uses a vision encoder based on ViT [10] architecture initialized with BEIT$_{base}$ [3] weights, pretrained on ImageNet-21k. The additional temporal attention modules are randomly initialized and added before spatial attention in each Transformer block as in [4]. As our text encoder, we use the first 9 layers of BERT$_{base}$ [9]. The multimodal fusion encoder is our previously described V2T-MF module built using the last 3 layers of the same BERT$_{base}$ model. Our final pretraining objective is the sum of VTC, VTM, MLM, and MVM losses. The hyperparameters are shown in Table 6. When doing multi-stage pretraining in Step 4 in the main draft, we set the initial learning rate of 5e-5 for stage 2 and 1e-6 for stage 3. Our model is implemented using PyTorch [33] with Mixed Precision Training [30] and Gradient Checkpointing [7].

Training Time. We train 2M and 5M corpus on $8\times$ RTX A5000 GPUs, which takes about 1 day and 1.8 days, respectively. For 17M and 25M, we train our model using $32\times$ A5000 GPUs, which takes 1.3 days and 3 days, respectively. For downstream tasks, the finetuning time ranges from 2-40 hours depending on the dataset size. The speed of A5000 is $0.99\times$ as V100 and $0.5\times$ as the A100 according to Lambda's benchmark[1].

[1] https://lambdalabs.com/gpu-benchmarks fp16, bert_base_squad

3. Dataset Descriptions

Pretraining. We pretrain our model on three corpora: C5M, C17M and C25M, which we describe below.

- **C5M (5M)**: WebVid-2M [2], and CC3M [34]. It contains a total of 5.44M image/video and text pairs.
- **C17M (17M)**: C5M, COCO [8], Visual Genome [18], SBU Captions [32], and CC12M [6]. It contains a total of 18.41M image/video and text pairs.
- **C25M (25M)**: C17M, and WebVid-10M [2] (excluding 2M videos from WebVid-2M as WebVid-10M is a superset of WebVid-2M). It contains a total of 25.91M image/video and text pairs.

Text-to-Video Retrieval. We evaluate our model on 3 spatially biased datasets MSR-VTT [39], DiDeMo [1], ActivityNet- Captions [17] and 2 temporally-heavy datasets SSv2-Template [19], SSv2-Label [19].

- **MSRVTT [39]** contains 10K YouTube videos with duration between 10-30 seconds and 200k captions. Following [2,41], we train on 9K videos and report results on 1K-A test set.
- **DiDeMo [1]** contains 10K Flicker videos with 41K captions. Following [19, 20, 24], we only keep the first 30 seconds of each video and evaluate paragraph-to-video retrieval, where all the descriptions for a video are concatenated to form a single query.
- **ActivityNet-Captions [5]** contains 20K YouTube videos with 100K captions. Following [19, 29], we

train on the train set with 10K videos and evaluate on the val set with 4.9K videos and evaluate paragraph-to-video retrieval.

- **SSv2-Template [19]** contains 169K videos for training and 2K videos for evaluation from dataset SSv2 [15]. The queries are 174 template (e.g., "Holding [something] next to [something]") in SSv2. In the 2K test set, each template has 12 videos.
- **SSv2-Label [19]** contains the same videos for train/test as in SSv2-Template except that the text queries are the annotated labels (e.g., "holding potato next to vicks vaporub bottle") in SSv2.

Video Question Answering. We evaluate on two open-ended QA datasets ActivityNet-QA, MSRVTT-QA and two multiple-choice QA dataset MSRVTT-MC, TVQA.

- **ActivityNet-QA** [42] contains 58K open-ended questions on 5.8K sampled videos from ActivityNet [17].
- **MSRVTT-QA** [38] contains 244K open-ended questions on 10K MSRVTT videos.
- **MSRVTT-MC** [41] contains 3K sampled videos with one multiple choice question for each video with 5 candidates. We evaluate the performance using the retrieval model finetuned on MSRVTT 7K training set.
- **TVQA** [21] contains 22K video clips and 153K multiple-choice questions focused on popular TV shows. We use the official train/val/test splits and reports results on the test set.

4. Additional Temporal Modeling Baselines

As discussed in the main draft, our first step is to extend our initial image transformer to video via a temporal modeling mechanism. Such a temporal modeling mechanism would enable training our model on multiple frames for more robust VidL spatiotemporal representation learning. For this part of our empirical study, we experiment with the following temporal modeling schemes using 4-frame inputs and pretrained on WebVid-2M [2]. Besides the four temporal modeling baselines (i.e., mean pooling (MP), late temporal attention (L-TA), temporal convolution (TC), and temporal attention (TA)) that we included in the main draft, we further study Temporal Attention via Prompts (TA-P) and Window Attention (WA). We describe each of these baselines in more detail below:

- **Temporal Attention via Prompts (TA-P).** Following, several previous methods [31, 40] we implement a baseline that uses temporal attention via prompt tokens. As shown in Figure 2, we first add m prompt tokens to each frame. Then, these prompt tokens attend to each other via temporal attention [4] to exchange frame-level information. Finally, all frame-level image tokens and prompt tokens for that frame attend to each other via spatial attention. Our TA-P scheme follows the same implementation as in [31].
- **Window Attention (WA).** Similar to Swin [25], the spatial-temporal tokens are divided into cuboids of size $T \times k \times k$, where T is the number of frames and k is the window size. WA is performed inside each cuboid. Similar to Temporal Attention, the WA is inserted before the spatial attention as in [4]. We experiment with $k = 2$ and $k = 7$. Larger k leads to an out-of-memory error.

Module	M	D	A	Avg.	Mem(GB)
Mean Pooling	49.4	53.7	46.4	50.1	**9.3**
Late Temp. Attn	50.3	54.3	46.0	50.6	10.3
Temp. Conv	53.0	58.2	52.7	54.6	10.3
Temp. Attn	53.7	**60.9**	55.6	56.7	11.4
Temp. Attn Promp.	49.5	52.7	46.6	49.9	10.3
Wind. Attn ($k = 2$)	**55.4**	59.0	56.2	56.9	12.5
Wind. Attn ($k = 7$)	54.6	59.9	**57.7**	**57.4**	18.1

Table 7. We study various temporal modeling schemes. M, D and A represents MSR-VTT, DiDeMo and ActivityNet-Captions. The accuracies are averaged R-{1,5,10}. GPU memory is measured with a batch size of 32 and gradient checkpointing enabled. Temporal Attention is the same as Window Attention with $k = 1$. We observe that a larger temporal modeling capacity leads to higher performance. However, Window Attention with large window size (i.e., $k = 7$) only has slight benefits (+0.6%) compared to Temporal Attention but a large increased GPU memory consumption. Thus, we use temporal attention for our subsequent experiments due to a favorable computational cost and accuracy balance. These experiments are conducted with 4-frame inputs, without a multimodal fusion encoder, and using the VTC loss as described in Step 1 of the main draft.

We also illustrate these attention mechanisms in Figure 2. Furthermore, for completeness, below, we also describe the four baselines included in the main draft of the paper.

- **Mean Pooling (MP).** In this variant, the visual encoder processes input frames independently and averages their frame-wise scores for the video-level score as in [29].
- **Late Temporal Attention (L-TA).** In this variant, we attach 2 Transformer layers to an image encoder, which then aggregates temporal information across all input frames.

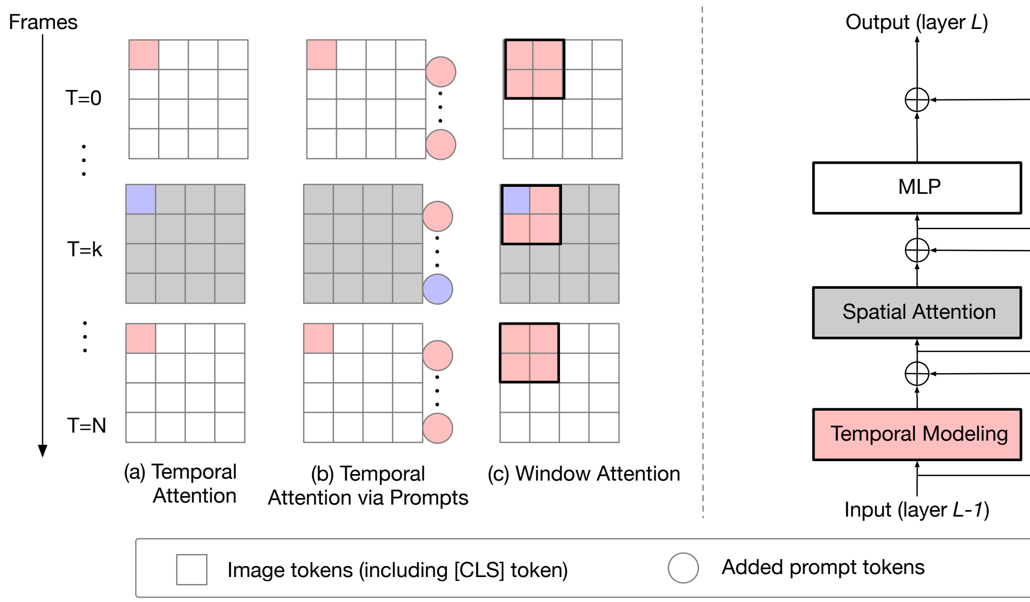

Figure 2. **Left**: Comparison of different attention mechanisms. The query token will first attend to tokens in other frames via temporal attention machnism and then attend to tokens in the same frames via spatial attention. In Temporal Attention via Prompts, only spatial attention is applied to image tokens. **Right**: The temporal modeling blocks are inserted before the spatial attention in each ViT layer.

- **Temporal Convolution (TC).** We insert a TC block before the spatial attention in each ViT layer. The TC block consists of a linear down-projection layer with hidden size 384, a depth-wise $3 \times 1 \times 1$ convolution as in [35], a ReLU activation, and a linear up-projection layer.
- **Temporal Attention (TA).** We insert a TA before spatial attention in each layer as in TimeSformer [4].

As shown in Table 7, Temporal Attention outperforms Temporal Convolution and Temporal Attention via Prompts by 2.1% and 6.8% respectively on averaged top-$\{1,5,10\}$ accuracy. Window Attention with window sizes of $k = 2$ and $k = 7$ outperforms Temporal Attention by 0.2% and 0.7% respectively. These results indicate that high temporal modeling capacity is important in VidL models. As Window Attention has $k\times$ the computational and memory cost and limited performance improvement compared with Temporal Attention, we choose Temporal Attention as our final temporal modeling blocks.

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017.

[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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021.

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.

[6] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021.

[7] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[8] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[11] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.

[12] 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.

[13] 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. *arXiv preprint arXiv:2209.01540*, 2022.

[14] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[15] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

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

[17] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017.

[18] 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. *International journal of computer vision*, 123(1):32–73, 2017.

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

[20] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021.

[21] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018.

[22] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

[23] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

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

[25] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[26] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022.

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[29] 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 and captioning. *Neurocomputing*, 508:293–304, 2022.

[30] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

[31] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022.

[32] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011.

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[34] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[35] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolu-

tional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[36] 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. *arXiv preprint arXiv:2203.07303*, 2022.

[37] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Luowei Zhou, Yucheng Zhao, Yujia Xie, Ce Liu, Yu-Gang Jiang, and Lu Yuan. Omnivl: One foundation model for image-language and video-language tasks. *arXiv preprint arXiv:2209.07526*, 2022.

[38] 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 *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653, 2017.

[39] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[40] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. *arXiv preprint arXiv:2209.06430*, 2022.

[41] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 471–487, 2018.

[42] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9127–9134, 2019.