# 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

## Abstract

*The last several years have witnessed remarkable progress in video-and-language (VidL) understanding. However, most modern VidL approaches use complex and specialized model architectures and sophisticated pretraining protocols, making the reproducibility, analysis and comparisons of these frameworks difficult. Hence, instead of proposing yet another new VidL model, this paper conducts a thorough empirical study demystifying the most important factors in the VidL model design. Among the factors that we investigate are (i) the spatiotemporal architecture design, (ii) the multimodal fusion schemes, (iii) the pretraining objectives, (iv) the choice of pretraining data, (v) pretraining and finetuning protocols, and (vi) dataset and model scaling. Our empirical study reveals that the most important design factors include: temporal modeling, video-to-text multimodal fusion, masked modeling objectives, and joint training on images and videos. Using these empirical insights, we then develop a step-by-step recipe, dubbed* VINDLU, *for effective VidL pretraining. Our final model trained using our recipe achieves comparable or better than state-of-the-art results on several VidL tasks without relying on external CLIP pretraining. In particular, on the text-to-video retrieval task, our approach obtains 61.2% on DiDeMo, and 55.0% on ActivityNet, outperforming current SOTA by 7.8% and 6.1% respectively. Furthermore, our model also obtains state-of-the-art video question-answering results on ActivityNet-QA, MSRVTT-QA, MSRVTT-MC and TVQA. Our code and pretrained models are publicly available at: https://github.com/klauscc/VindLU.*

## 1. Introduction

Fueled by the growing availability of video-and-text data [2,8,9,24,41,43,48] and advances in the Transformer model design [12,54], the last few years have witnessed incredible progress in video-and-language (VidL) understanding [26,31,40,64,75,80]. Since the initial transformer-based models for VidL, such as ClipBERT [26], the text-to-video retrieval accuracy has improved from 22.0%, 22.4%, and 21.3% on MSR-VTT [65], DiDeMo [1], and ActivityNet [23] to $>$ 45% R@1 accuracy on all three of these datasets, thus, marking an extraordinary relative improvement of more than 100% in less than 2 years.

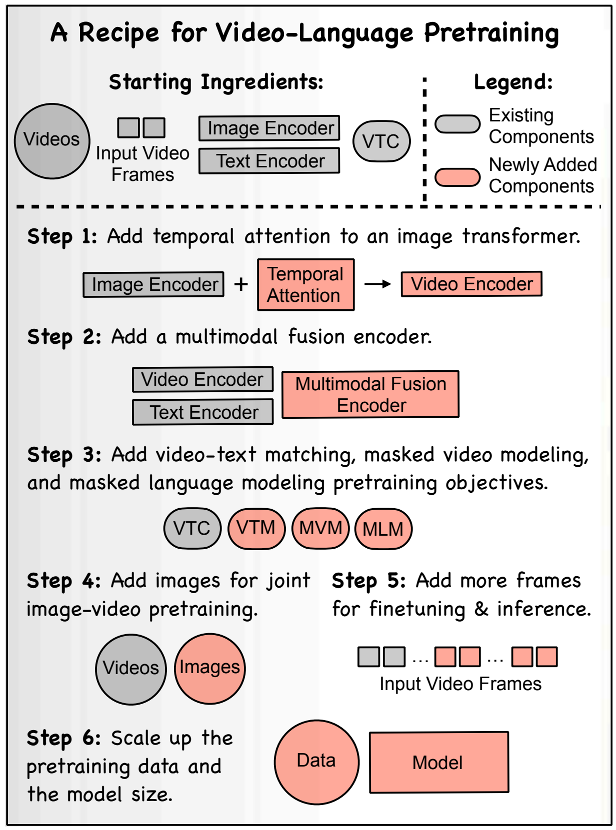


Figure 1. We present a recipe for effective video-language pretraining. Our recipe starts with image and text transformer encoders trained on video-text pairs using a contrastive objective (VTC). We then progressively add more components to our framework while also studying the importance of each component along the way. Our final recipe includes the steps for (1) adding temporal attention, (2) injecting a multimodal fusion encoder, (3) incorporating masked modeling pretraining objectives, (4) jointly training on images and videos, (5) using more frames during fine-tuning and inference, and lastly, (6) scaling up the data and the model.

At the same time, the model architectures and pretraining/finetuning protocols used by modern VidL approaches

| Method | Model Design | | | Pretraining Data | | | #Frames | | |
|---|---|---|---|---|---|---|---|---|---|
| | Temporal Modeling | Multimodal Fusion | Pretraining Objectives | Dataset | Size | Modality | PT | FT | Eval |
| UniVL [39] | Joint Att. [5] | 2-layer TR | VTC+VTM+MLM+MFM+LM | HT | 136M | V | 48 | 48 | 48 |
| VideoCLIP [64] | 1D-Conv+TR | ✗ | VTC | HT | 136M | V | 32 | 32 | 32 |
| ClipBert [26] | Mean Pooling | BERT | MLM+VTM | COCO+VG | 0.2M | I | 1 | 16 | 16 |
| Frozen [2] | Temp. Attn [5] | ✗ | ITC | C5M | 5M | I+V | $1 \rightarrow 4$ | 4 | 4 |
| MERLOT [75] | Joint Attn | RoBERTa | VTC+MLM+FOM | YT | 180M | V | 16 | 16 | 16 |
| VIOLET [16] | Window Attn [37] | BERT | VTC+VTM+MLM+MVM | YT+C5M | 185M | I+V | 4 | 5 | 5 |
| MV-GPT [47] | Joint Attn | 2-layer TR | MLM+LM | HT | 136M | V | - | - | - |
| ALL-in-one [55] | Token Rolling [55] | ViT | VTC+VTM+MLM | HT+W2 | 172M | V | 3 | 3 | 9 |
| Singularity [25] | Late Temp. Attn | 3-layer TR | VTC+VTM+MLM | C17M | 17M | I+V | $1 \rightarrow 4$ | 4 | 12 |
| LAVENDER [32] | Window Attn [37] | BERT | MLM | C17M+IN | 30M | I+V | 4 | 5 | 5 |
| OmniVL [57] | Temp. Attn | 2×BERT | VTC+VTM+LM | C17M | 17M | I+V | $1 \rightarrow 8$ | 8 | 8 |
| ATP [6] | ✗ | ✗ | VTC | CLIP | 400M | I | 1 | 16 | 16 |
| CLIP4Clip [40] | Late TR | ✗ | VTC | CLIP | 400M | I | 1 | 12 | 12 |
| ECLIPSE [34] | Late TR | ✗ | VTC | CLIP | 400M | I+A | 1 | 32 | 32 |
| CLIP2TV [18] | CLIP | 4-layer TR | VTC+VTM | CLIP | 400M | I | 1 | 12 | 12 |
| CLIP-Hitchhiker [3] | Late Attn | ✗ | VTC | CLIP | 400M | I | 1 | 16 | 120 |
| CLIP-ViP [66] | Prompt Attn [66] | ✗ | VTC | CLIP | 500M | I+V | $1 \rightarrow 12$ | 12 | 12 |

**TR**: Transformer; **Late**: Late fusion; **Attn**: Attention. **V**: Video; **I**: Image; **A**: Audio; $1 \rightarrow 4$: 1 frame for stage-1 training and 4 frames for stage-2. **VTC**: Video-text contrastive; **VTM**: Video-text matching; **MLM**: Masked language modeling; **MFM**: Masked frame modeling; **LM**: Language modeling. **HT**: HowTo100M [41]; **C5M, C17M**: see supplementary; **YT**: YT-Temporal [75]; **W2**: WebVid-2M [2]; **COCO**: [33], **VG**: Visual Genome [24]; **IN**: An internal dataset.

Table 1. An overview of the existing VidL methods. Significant differences exist among these methods, making it challenging to reproduce, analyze and compare these methods. This motivates us to answer the question "What are the key steps to build a highly performant VidL framework" by investigating various components in the VidL framework design.

have become significantly more complex and specialized over the last several years. As a result, it is increasingly difficult to reproduce, analyze and compare most recent VidL frameworks. For example, several recent approaches [25, 32, 66] propose new architectures, new initialization strategies, pretraining objectives, pretraining datasets, and optimization protocols. Due to the large computational cost of ablating all these factors, it is difficult to understand which components are critical to the success of the proposed frameworks. Similarly, the key success factors of many other recent VidL approaches [6, 16, 32, 57] are also often obfuscated, which hinders future research.

In Table 1, we illustrate the complexity of modern VidL frameworks by dissecting them along multiple dimensions, including temporal modeling schemes, multimodal fusion modules, pretraining objectives, the source of the pretraining data, and the number of frames for pretraining, finetuning and inference. Based on this analysis, we observe that there exist significant differences among these VidL methods. Unfortunately, it's not clear which differences are important for the overall VidL performance and which are not.

The recent METER [13] work studies a subset of these components in the context of image-language modeling. However, their analysis is limited to images and, thus, ignores various aspects related to video modeling, such as spatiotemporal architecture design, video pretraining objectives, video pretraining data, and video-specific finetuning/evaluation protocols such as the number of frames. As we will show in our experimental section, many of the findings presented in the image-based studies [13] do not hold for video. Beyond image-based analysis, we note that the concurrent work in [17] conducts an empirical study of VidL transformers. However, unlike our work, which covers a broad range of VidL design factors, their analysis is focused predominantly on masked visual modeling objectives, which we also study in this work.

Our main objective in this work is to answer the question "What are the key steps needed to build a highly performant VidL framework?" To do this, we conduct a thorough empirical study that demystifies the importance of various VidL design choices and ultimately leads to a VidL framework that achieves state-of-the-art results on various VidL benchmarks. Using our empirical insights, we then develop a step-by-step recipe for effective VidL pretraining. Our recipe, dubbed VINDLU (VIdeo aND Language Understanding), starts from a standard Vision Transformer (ViT) [12] and uses a simple progressive expansion scheme where at each step, we investigate a particular aspect of VidL framework design (e.g., architecture, pretraining objective, pretraining data, etc.), and choose the best performing option. In particular, we study the following VidL design components: (i) the spatiotemporal architecture design,

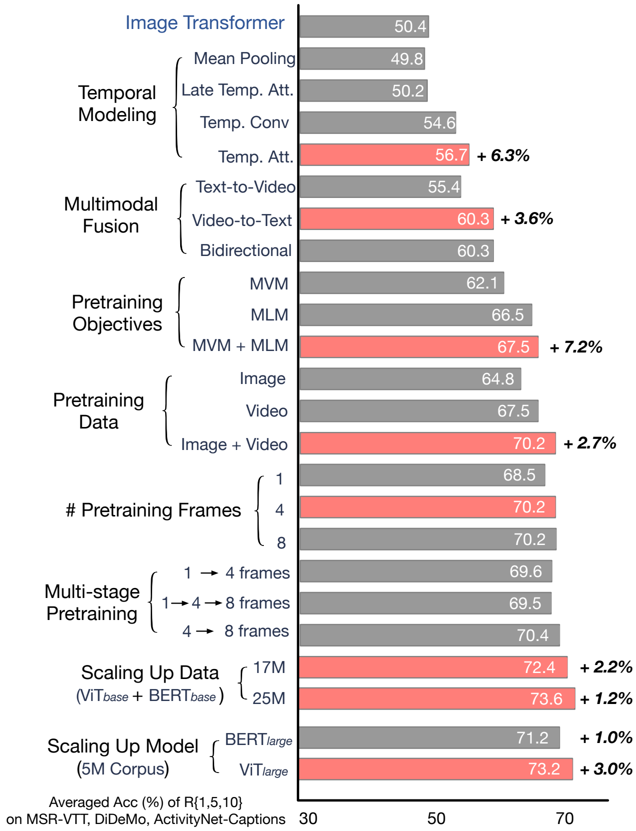


Figure 2. We progressively expand an image transformer baseline (e.g., ViT) to a performant video-and-language (VidL) model. We do so by investigating the importance of many VidL design choices such as (i) temporal modeling, (ii) multimodal fusion modules, (iii) pretraining objectives, (iv) the source of the pretraining data, (v) the number of pre-training frames, (vi) multi-stage pretraining, and (vii) scaling of the data and model. Each bar depicts an average text-to-video retrieval Recall@1,5,10 accuracy across MSR-VTT [65], DiDeMo [65], ActivityNet [23]. The red bars denote the best-performing design choice in each subgroup. Our final VidL framework, dubbed VINDLU, outperforms our initial image Transformer baseline by **23.2%**. The figure was inspired by [36].

(ii) the multimodal fusion schemes, (iii) the pretraining objectives, (iv) the source of the pretraining data, (v) fine-tuning/inference protocols, and (vi) scaling of the data and model. We present our recipe in Fig. 1.

The key findings of our empirical study include:

- Contrary to the conclusions of several prior works [6, 25] that a single frame is sufficient for VidL modeling, we discover that temporal modeling using multiple frames leads to a significant improvement over the spatial-only baselines (**+6%** averaged video retrieval accuracy on MSR-VTT, DiDeMo, and ActivityNet).
- Multimodal fusion module incorporating video features into text is critical for good VidL performance (**+3.6%**). Conversely, adding text features to the video representation is not useful.
- Masked language modeling objective significantly improves performance (**+6.2%**) while masked video modeling objective brings an additional **+1%** improvement.
- Pretraining jointly on images and videos is beneficial (**+2.7%**). Also, contrary to prior methods [2, 57], we find multi-stage training unnecessary.
- Pretraining with a small number of frames (e.g., 4) is sufficient and it can significantly reduce the computational cost of large-scale pretraining. Pretraining with more frames does not lead to a substantial performance boost.
- Compared to many recent CLIP-based [45] VidL approaches [3, 40, 66], our recipe achieves comparable or even better performance with **20**$\times$ less pretraining data.

Our final model, trained using our VINDLU recipe, achieves state-of-the-art results on several VidL benchmarks. Specifically, on the video retrieval task, our method achieves 46.5%, 61.2%, 55.0% R@1 accuracy on MSR-VTT, DiDeMo, and ActivityNet outperforming the state-of-the-art by **7.8%** and **6.1%** on the latter two datasets. Also, our approach obtains state-of-the-art video question-answering results on ActivityNet-QA, MSRVTT-QA, MSRVTT-MC and TVQA, where we achieve top-1 accuracy of 44.7%, 44.6%, 97.1%, and 79.0% respectively.

We want to make it clear that, in this paper, we do not claim technical novelty behind any of the individual design choices (i.e., different subsets of these design choices were already used by prior VidL methods as shown in Table 1). Instead, our main contribution, which we believe might be equally if not more important than proposing yet another specialized or obfuscated VidL model, is to investigate these components collectively and validate their importance. We also do not claim superiority over previous methods (despite better results). Due to the implementation complexities of each method, fair and complete comparisons are difficult and not our intent. Instead, we hope that our recipe for building an effective VidL framework will provide useful insights for future research on VidL understanding. To enable the VidL community to build on our work, we release our code and pretrained models.

## 2. Related Work

**Image-and-Language Pretraining.** Recent years have witnessed remarkable progress in image-and-language pretraining [7, 10, 20, 22, 29, 38, 45, 49, 51, 59–61, 69, 70, 74, 76–79]. However, most modern methods such as ViLBERT [38], UNITER [10], CoCa [71], LEMON [20], BEiT-3 [61] employ complex transformer architectures and pretraining objectives. Thus, it is difficult to decipher which components are critical for good performance. A recent empirical study on image-language modeling METER [13] studies a variety of components. However, since their analysis is done exclusively on images, it is unclear whether these findings generalize to video. In comparison, our work thoroughly investigates various video-specific design choices for effective video-language pretraining.

**Video-and-Language Pretraining.** In recent years, the large-scale VidL pretraining [6,16,26,32,57,58] has shown strong transfer learning ability to downstream VidL tasks such as text-to-video retrieval [1,23,26,35,40,65,72], video question answering [63, 72, 73], video captioning [21, 23, 50, 56, 81], etc. Several methods [3, 18, 40, 66] achieve impressive results by building on the popular image-language pretrained model CLIP [45]. Additionally, several recent approaches [25, 32, 57] propose more sophisticated VidL frameworks to achieve comparable performance as CLIP-based methods without large-scale CLIP pretraining. However, with the impressive results, these methods also require more complex architectures and specialized video pretraining protocols (as shown in Table 1). The complexity of these frameworks and the large computational cost of VidL pretraining makes it challenging to decipher which VidL framework components are truly needed for good performance. Moreover, unlike in the image-language domain, there are few empirical studies investigating various VidL design components collectively. For instance, the concurrent work of Fu [17] only studies masked video modeling pretraining objectives and is based on a slightly older VIOLET [16] method. Furthermore, the recent works [6,25] focus predominantly on spatial biases in modern VidL benchmarks. In contrast to these approaches, our work investigates the importance of various factors in VidL framework design. We then use our empirical insights to provide a detailed step-by-step recipe for effective VidL pretraining.

# 3. A Recipe for Video-Language Pretraining

In this section, we describe our recipe for video-and-language (VidL) pretraining. We begin with a standard image transformer (e.g., ViT [12]) and progressively expand it to a model that achieves state-of-the-art results on various VidL datasets and tasks. At each step of our recipe, we study how various design choices affect VidL performance. In particular, we are interested in answering the following questions about the VidL pretraining design:

- Does a VidL model need temporal modeling, especially since most VidL benchmarks are spatially biased [6,25]? If so, what is the best temporal modeling scheme?
- What is the most effective way to do multimodal fusion? Some approaches [16, 32, 55] use bidirectional while others [25, 57] employ unidirectional multimodal fusion modules. Which of these schemes works the best?
- Which pretraining objectives are most useful for VidL representation learning? Prior methods use video-text contrastive (VTC) [28], video-text matching (VTM) [28, 31, 39], masked-language-modeling (MLM) [11], or masked-video-modeling (MVM) [52]. Are all of these objectives needed for the best performance?
- What pretraining data is most useful for training VidL models (e.g., video-only or images and videos)? Is it necessary to use curriculum learning [2, 55, 57] or is single-stage pretraining sufficient?
- How many frames are needed for pretraining, fine-tuning, and inference? Several approaches [6, 25] claimed that single frame pretraining is sufficient while others [57,66] pretrained their models with 8 or even more frames. Should we finetune the pretrained VidL models using the same number of frames as during pretraining or is it helpful to use more frames during fine-tuning and inference?

Motivated by these questions, we next present our recipe while also studying these questions in more detail.

## Step 0: Starting Ingredients

**Image Transformer Baseline.** We start with a standard ViT-B/16 [12] transformer trained on single frames of WebVid-2M [2]. We use BERT [11] as our text encoder for all experiments. Formally, given the paired video and text input $(v, t)$, The image transformer randomly selects a single frame from the video as input to extract the video embeddings. A text encoder encodes the text $t$ to extract the text embeddings. We then use a video-text contrastive (VTC) loss to maximize the agreement between the paired video and text embeddings as in [2, 45]. Following [25], we use BEiT [4] initialization for our image transformer, whereas the text encoder is initialized with BERT$_{base}$.

**Experimental Setup.** As our initial pretraining data, we use WebVid-2M [2] unless noted otherwise. We then finetune and evaluate our pretrained model on the three popular text-to-video retrieval datasets: MSR-VTT [65], DiDeMo [1], and ActivityNet-Captions [23], which include short and long videos. We report the averaged top-1, top-5, and top-10 text-to-video retrieval accuracies across these datasets as our evaluation metric. As shown in Fig. 2, our Image Transformer baseline achieves an average accuracy of **50.4%**.

Over the next several subsections, we progressively expand this baseline by adding more components of increasing complexity. In particular, we start by incorporating (i) temporal modeling blocks, (ii) a multimodal fusion encoder, and (iii) additional pretraining objectives. Afterward, we investigate the choice for the (iv) pretraining data, (v) fine-tuning and inference protocols, and (vi) dataset and model scaling schemes. We would like to note that due to the large computational cost, we cannot ablate the order of the steps in our recipe. Thus, the order of the steps is primarily determined by the computational cost (i.e., the steps that can be implemented most efficiently are studied first then, moving to the more computationally costly steps).

## Step 1: Temporal Modeling

In the first step of our recipe, we extend our initial image transformer to video via a temporal modeling mechanism, which enables training our model on multiple frames. We experiment with several temporal modeling schemes:

- **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 [40].
- **Late Temporal Attention (L-TA).** Following [25,40,42] we use a late temporal modeling scheme by attaching 2 Transformer layers to an image encoder, which then aggregates temporal information across all input frames.
- **Temporal Convolution (TC).** Many prior methods [14, 44, 62] used 3D convolutions for temporal modeling. To validate its effectiveness, we inject 3D convolution [53] before the spatial attention to each Transformer Layer.
- **Temporal Attention (TA).** Inspired by TimeSformer [5], we experiment with divided space-time attention, which we insert before spatial attention as in [5].

As shown in the upper part of Fig. 2 and the Table below, the temporal modeling capability is critical for good VidL performance. This is indicated by a $+\mathbf{6.3}\%$ accuracy boost of our temporal attention variant (TA) over the spatial-only baseline. We also observe that late temporal modeling (L-TA) has nearly no effect. We conjecture that this is due to the limited temporal modeling capacity (*i.e.*, only two layers) and the lack of temporal fusion in the early layers. Lastly, our results suggest that TA outperforms TC by **2.1%**, which might indicate that long-range temporal attention is more useful than local 3D convolutions.

| | Mean Pooling | L-TA | TC | TA |
|---|---|---|---|---|
| acc.(%) | 49.8 | 50.2 | 54.6 | **56.7** |

Interestingly, we note that our findings contradict the conclusions of several recent methods [6, 25], claiming that temporal modeling is not needed for many VidL tasks. We hypothesize that even on the spatially-biased datasets, temporal modeling is useful for resolving spatial ambiguities caused by appearance variations across different frames.

*Takeaway #1: We adopt Temporal Attention (TA) as our temporal modeling mechanism and pretrain our model with 4-frame inputs unless otherwise noted.*

## Step 2: Multimodal Fusion Encoder

Building on the model from Step 1, we next analyze the role of multimodal fusion modules. The multimodal fusion encoder aims to fuse multimodal cues from video and language for a more discriminative VidL feature representation. As shown in Fig. 3, we experiment with several variants of multi-modal fusion encoders:

- **Video-to-Text Multimodal Fusion (V2T-MF).** As illustrated in Fig. 3a, V2T-MF injects relevant video cues into the textual features using Cross-Attention. For a fair comparison with previous baselines [2,25], we do not add any extra layers but instead re-purpose the last $m$ layer of our text encoder for V2T fusion. Specifically, a cross-attention operation is inserted into each of the $m$ last layers in the text encoder between Self-Attention and MLP. This scheme was also previously used by [25, 57].
- **Text-to-Video Multimodal Fusion (T2V-MF).** Similar to V2T-MF, we build T2V-MF (Fig. 3b) by re-purposing the last $m$ layers of the vision encoder and using cross-attention to incorporate text cues into the video features.
- **Bidirectional Multimodal Fusion (B-MF).** Prior approaches [16, 32, 55, 75] feed the concatenated visual and textual features to a $m$-layer Transformer. However, this is often computationally infeasible in the video domain due to many input frames. Instead, we implement B-MF by combining T2V-MF and V2T-MF.

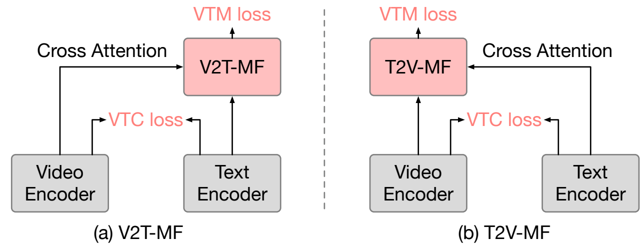


Figure 3. An illustration of (a) video-to-text (V2T-MF), and (b) text-to-video (T2V-MF) multimodal fusion schemes. The video-text matching (VTM) loss is attached to the multimodal fusion encoder, whereas video-text contrastive (VTC) loss is added to the video and text encoders.

To train each variant, we add the video-text matching (VTM) loss (see Sec. 3) as in [16,55,57]. In the table below and Figure 2, we report that the V2T-MF scheme performs the best (i.e., **+3.6%** improvement). Surprisingly, the reverse T2V-MF scheme substantially decreases performance (**-1.3%**). We conjecture that predicting the matching video-text pairs using a pretrained language rather than a visual representation is easier. We also note that the B-MF scheme yields no improvement compared to V2T-MF.

| | w/o. MF | T2V-MF | V2T-MF | B-MF |
|---|---|---|---|---|
| acc.(%) | 56.7 | 55.4 | **60.3** | **60.3** |

*Takeaway #2: For our remaining experiments, we use V2T-MF as our multimodal fusion encoder.*

## Step 3: Pretraining Objectives

Building on Step 2, we next study the following pretraining objectives:

- **Visual-Text Contrastive Learning (VTC).** VTC aims to learn independent representations for video and text by maximizing the agreement between positive *(visual, text)* pairs while minimizing the agreement between negative pairs. Note that this objective is already used in previous steps, and thus, not included in Figure 2.

- **Visual-Text Matching (VTM)**. VTM objective is implemented as a standard cross-entropy loss that encourages a VidL model to produce binary predictions indicating whether a given video-text pair matches. Following [30], we attach this loss to our multimodal fusion encoder and use hard negative mining during training as in [25]. The VTM objective is already used in Step 2 (i.e., the multimodal fusion step) and thus, not included in Figure 2.
- **Masked Language Modeling (MLM).** MLM objective aims to predict the masked words by leveraging information from both visual and textual features. We mask 50% text tokens using the same masking strategy as in BERT [80] and attach a linear layer to our multimodal fusion encoder (T2V-MF) to predict the masked words.
- **Masked Video Modeling (MVM)**. The MVM objective aims to recover the masked video tokens [15, 19, 37, 52]. To implement MVM, we apply a linear layer on the vision encoder and predict the masked tokens as in [46].

In the Table below and Fig. 2, we report that the MLM pretraining objective leads to a substantial boost in performance (**+6.2%**). Furthermore, adding MVM loss further improves the accuracy by **1%**. However, adding the MVM objective slows the training by about 40% (due to additional forward and backward passes). Thus, to speed up the training, we don't use MVM loss in our remaining experiments.

| objectives | acc.(%) |
|---|---|
| VTC (Step 1) | 56.7 |
| VTC+VTM (Step 2) | 60.3 |
| VTC+VTM+MLM | 66.5 |
| VTC+VTM+MLM+MVM | **67.5** |

*Takeaway #3: For the remaining experiments, we use VTC, VTM, MLM as our pretraining objectives.*

## Step 4: Pretraining Data

In this section, we analyze the effect of (i) the pretraining data, and (ii) pretraining protocols.

**Datasets.** Recent methods [2, 16] suggest that jointly pretraining on images and videos leads to better performance. To investigate this, we consider an additional image-based CC3M [48] consisting of 3M image-text pairs. Specifically, we experiment with pretraining our model on the (i) image-only (CC3M), (ii) video-only (WebVid2M), and (iii) joint image and video (CC3M + WebVid2M) datasets. When pretraining on images, we replace our previously introduced temporal attention module with an identity connection. As shown in the Table below and Fig. 2, training on videos is more beneficial than training on images (**+2.7%**). Furthermore, jointly pretraining on images and videos leads to an additional **2.7%** boost, which suggests that a stronger spatial representation is useful for VidL modeling.

| | Images | Videos | Images+Videos |
|---|---|---|---|
| acc.(%) | 64.8 | 67.5 | **70.2** |

**The Number of Input Frames for Pretraining.** Prior approaches [2, 16, 57, 75] use a different number of input frames for pretraining (i.e., from 1 to 16). Thus, we next study how many frames are needed for effective VidL pretraining. From the Table below and Fig. 2, we observe that multi-frame pretraining using 4 frames leads to **1.7%** improvement compared to a single-frame pretraining. However, we also observe that the performance saturates with 4-frame inputs while the computational cost of pretraining with more frames increases significantly, i.e., pre-training with 4 frames is **2.5**$\times$ faster than pretraining with 16 frames.

| | 1 frame | 4 frames | 8 frames | 16 frames |
|---|---|---|---|---|
| acc.(%) | 68.5 | **70.2** | **70.2** | **70.2** |
| speedup | **4.6**$\times$ | 2.5$\times$ | 1.7$\times$ | 1$\times$ |

**Multi-stage Curriculum Pretraining.** Lastly, we validate the necessity of multi-stage curriculum pretraining, which was used in several prior VidL approaches [2, 57]. Specifically, we experiment with two different pretraining protocols: (i) a two-stage pretraining that first trains a model for 10 epochs using single frames, and then for 5 additional epochs using 4-frame inputs, and (ii) a three-stage pretraining that builds on (i) by adding a third stage where the model is trained for additional 3 epochs using 8-frame inputs. Our results in the Table below and Figure 2, indicate that multi-stage pretraining does not lead to any significant performance boost, contrary to the findings of prior approaches [2, 57]. We believe that this happens because prior approaches [2, 57] train their model for only several epochs at each stage, whereas we train it until convergence. We also note that compared to the 4-frame one-stage pretraining, the two-stage $1 \rightarrow 4$ has a comparable pretraining cost as the latter model is trained for more epochs.

| frames | 4 | $1 \rightarrow 4$ | $1 \rightarrow 4 \rightarrow 8$ | $4 \rightarrow 8$ |
|---|---|---|---|---|
| acc.(%) | 70.2 | 69.6 | 69.5 | **70.4** |
| speedup | **1.7**$\times$ | **1.7**$\times$ | 1.2$\times$ | 1$\times$ |

*Takeaway #4: We adopt a single-stage pretraining on joint image and video datasets while using 4-frame inputs.*

## Step 5: Finetuning & Inference

Existing methods typically use the same number of frames either between pretraining and finetuning [2, 25, 75] or finetuning and inference [16, 57, 75]. Here, we study using a different number of frames at different phases.

**Finetuning.** We experiment with finetuning our 4-frame pretrained model with $K = 1, 4, 8, 12, 24, 32$-frame inputs while using $M$ frames during inference. We use $M = 12$ for all $K \leq 12$ and $M = K$ for $K > 12$. Based on the results in the Table below, we observe that while finetuning with more frames leads to higher accuracy (**70.5%**) the performance saturates with about 12 frames. We also note that finetuning with a single-frame input is **22.4**$\times$ faster than with 32 frames but has a **5%** lower accuracy. On the other hand, finetuning with 12 frames yields only **0.3%** lower accuracy but **2.6**× speedup compared to finetuning with 32 frames. Therefore, due to the favorable accuracy-cost trade-off, we finetune most of our models with 12-frame inputs.

| # frames | 1 | 4 | 8 | 12 | 24 | 32 |
|---|---|---|---|---|---|---|
| acc.(%) | 65.5 | 68.1 | 69.2 | 70.2 | 70.1 | **70.5** |
| speedup | **22.4**× | 7.1× | 3.9× | 2.6× | 1.5× | 1.0× |

**Inference.** Next, we experiment with 12, 24, 32, 64 frames for testing our 4-frame pretrained and 12-frame finetuned model. We report the averaged accuracies on DiDeMo (D) / ActivityNet (A), which contain longer videos. Using more frames for inference helps, but the accuracy saturates quickly, and the inference cost becomes large.

| # frames | 12 | 24 | 32 | 64 |
|---|---|---|---|---|
| D/A acc.(%) | 73.4/70.4 | 73.0/72.1 | 72.7/72.6 | **73.8/72.8** |
| speedup | **10.6**× | 3.1× | 2.1× | 1× |

*Takeaway #5: Considering the trade-off between computational cost and accuracy, we use 12 frames for finetuning and inference on all datasets except ActivityNet. On ActivityNet, we use 12 and 32 frames for finetuning and inference.*

### Step 6: Scaling Up

Lastly, we scale up the pretraining data and the model.

**Pretraining Data.** For the pre-training data, we experiment with (a) adding 12M images from CC12M for a **17M Corpus**, and (b) additional 10M videos from WebVid10M for a **25M Corpus**. The results in the Table below and in Fig. 2 indicate that scaling our corpus from $5M \rightarrow 17M$ improves the performance by **2.2%**. Furthermore, scaling the corpus from $17M \rightarrow 25M$ leads to an additional boost of **1.2%**.

| # corpus | 5M | 17M | 25M |
|---|---|---|---|
| acc.(%) | 70.2 | 72.4 | **73.6** |

**Model Size.** We also experiment with scaling the video encoder (ViT$_{base}$ → ViT$_{large}$) or text encoder (BERT$_{base}$ → BERT$_{large}$). Due to the large computational cost, we only conduct these experiments on the 5M corpus. We report that scaling the vision encoder brings larger improvement ( **+3.0%**) than scaling the text encoder (**+1.0%**).

| encoders | base | ViT$_{large}$ | BERT$_{large}$ |
|---|---|---|---|
| acc.(%) | 70.2 | **73.2** | 71.2 |

*Final Takeaway: Our final scaled-up* VINDLU *model improves the initial image transformer baseline by **23.2%**.*

## 4. Experimental Results

We validate our VINDLU recipe on two mainstream VidL tasks. See implementation details and dataset descriptions in the supplementary material.

**Text-to-Video Retrieval.** We compare our results with existing methods on three spatially-biased datasets MSR-VTT, DiDeMo, and ActivityNet and two temporally-heavy datasets, SSv2-label, and SSv2-template as shown in Tab. 2 and Tab. 3 respectively. Our method outperforms previous methods by a large margin on multiple datasets, achieving averaged accuracies of 79.3% (**+5.6%**), 75.4% (**+4.7%**), 84.6% (**+4.6%**) on DiDeMo, ActivityNet-Captions and SSv2 respectively. Our results on MSR-VTT are worse (**66.5%** vs. **68.6%**) than OmniVL [57] but our pretraining framework is significantly cheaper (i.e., **82** vs. **169** V100 GPU days). We also note that our method is significantly cheaper than other top-performing approaches including LAVENDER [32], All-in-one [55], and CLIP-ViP [66] (**82** vs. **640, 448, 984** V100 GPU days for pretraining respectively). Additionally, our cheapest VINDLU variant requires only **15** V100 GPU days for pre-training, which is the second cheapest model among all listed approaches, and it still achieves competitive results on all three benchmarks. Furthermore, compared to the other leading VidL approaches such as OmniVL and Singularity, which rely on a multi-stage curriculum pretraining, our framework is simpler since it can be trained in a single stage. We also include the results of our scaled up variant VINDLU-L that uses ViT$_{large}$ as its video encoder, and report that it achieves 74.5% averaged retrieval accuracy, thus, outperforming all other approaches. Lastly, our results on the SSv2 dataset in Table 3 indicate that VINDLU performs well not only on spatially-biased datasets but also on temporally-heavy datasets, which require sophisticated temporal modeling capabilities. For fairer comparisons, we de-emphasize CLIP-based methods since they use a lot more pre-training data.

**Video Question-Answering.** In Table 4, we also present our results for the video question-answering task on ActivityNet-QA [73], MSRVTT-QA [63], MSRVTT-MC [72] and TVQA [27]. Our results indicate that compared to prior state-of-the-art approaches, VINDLU achieves competitive results across all four of these datasets. In particular, our method outperforms existing approaches by **0.6%** on ActivityNet-QA, **0.3%** on MSRVTT-QA, **3.4%** on MSRVTT-MC and **0.3%** on TVQA. For fair comparison, we de-emphasize FrozenBiLM [68], since it is a lot larger than our model (1.2B vs. 201M parameters) and uses a lot more pretraining data (400M vs. 25M).

## 5. Conclusion

In this work, we demystify the importance of various components used in modern VidL framework design. Throughout our empirical study, we find that temporal modeling, multimodal fusion, masked modeling pretraining objectives, and joint training on images and videos are critical for good performance on the downstream VidL under-

| Method | Pretrain | | | MSRVTT | | | | DiDeMo | | | | ActivityNet-Captions | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | #Data | #Frames | Time | R1 | R5 | R10 | Avg | R1 | R5 | R10 | Avg | R1 | R5 | R10 | Avg | |
| ClipBERT [26] | 5.4M | 1 | 32 | 22.0 | 46.8 | 59.9 | 42.9 | 20.4 | 48.0 | 60.8 | 43.1 | 21.3 | 49.0 | 63.5 | 44.6 | 43.5 |
| VideoCLIP [64] | 136M | 960 | 8 | 30.9 | 55.4 | 66.8 | 51.0 | - | - | - | - | - | - | - | - | - |
| Frozen [2] | 5M | $1 \rightarrow 4$ | $35^*$ | 31.0 | 59.5 | 70.5 | 53.7 | 34.6 | 65.0 | 74.7 | 58.1 | - | - | - | - | - |
| ALPRO [28] | 5M | 8 | $24^*$ | 33.9 | 60.7 | 73.2 | 55.9 | 35.9 | 67.5 | 78.8 | 60.7 | - | - | - | - | - |
| VIOLET [16] | 138M | 4 | 83 | 34.5 | 63.0 | 73.4 | 57.0 | 32.6 | 62.8 | 74.7 | 56.7 | - | - | - | - | - |
| All-in-one [55] | 138M | 3 | 448 | 37.9 | 68.1 | 77.1 | 61.0 | 32.7 | 61.4 | 73.5 | 55.9 | 22.4 | 53.7 | 67.7 | 47.9 | 54.9 |
| LAVENDER [32] | 30M | 4 | 640 | 40.7 | 66.9 | 77.6 | 61.7 | 53.4 | 78.6 | 85.3 | 72.4 | - | - | - | - | - |
| Singularity [25] | 17M | $1 \rightarrow 4$ | 29 | 42.7 | 69.5 | 78.1 | 63.4 | 53.1 | 79.9 | 88.1 | 73.7 | 48.9 | 77.0 | 86.3 | 70.7 | 69.3 |
| OmniVL [57] | 17M | $1 \rightarrow 8$ | $169^*$ | 47.8 | **74.2** | **83.8** | **68.6** | 52.4 | 79.5 | 85.4 | 72.4 | - | - | - | - | - |
| CLIP4Clip [40] | 400M | 1 | $768^*$ | 44.5 | 71.4 | 81.6 | 65.8 | 42.8 | 68.5 | 79.2 | 63.5 | 40.5 | 72.4 | 83.4 | 65.4 | 64.9 |
| ECLIPSE [34] | 400M | 1 | $768^*$ | - | - | - | – | 44.2 | - | - | - | 45.3 | 75.7 | 86.2 | 69.1 | - |
| CLIP-Hhiker [3] | 400M | 1 | $768^*$ | 47.7 | 74.1 | 82.9 | 68.6 | - | - | - | - | 44.0 | 74.9 | 86.1 | 68.3 | - |
| CLIP-ViP [66] | 500M | $1 \rightarrow 12$ | $984^*$ | 54.2 | 77.2 | 84.8 | 72.1 | 50.5 | 78.4 | 87.1 | 72.0 | 53.4 | 81.4 | 90.0 | 74.9 | 73.0 |
| VINDLU | 5M | 4 | 15 | 43.8 | 70.3 | 79.5 | 64.5 | 54.6 | 81.3 | 89.0 | 75.0 | 51.1 | 79.2 | 88.4 | 72.9 | 70.8 |
| | 17M | | 38 | 45.3 | 69.9 | 79.6 | 64.9 | 59.2 | 84.1 | 89.5 | 77.6 | 54.4 | 80.7 | 89.0 | 74.7 | 72.4 |
| | 25M | | 82 | 46.5 | 71.5 | 80.4 | 66.1 | **61.2** | 85.8 | 91.0 | **79.3** | 55.0 | 81.4 | 89.7 | 75.4 | 73.6 |
| VINDLU-L | 25M | 4 | 178 | **48.8** | 72.4 | 82.2 | 67.8 | 59.8 | **86.6** | **91.5** | **79.3** | **55.9** | **82.3** | **90.9** | **76.4** | **74.5** |

Table 2. Comparison to the state-of-the-art text-to-video retrieval methods on MSRVTT, DiDeMo and AcitivityNet-Captions. Pretraining time is measured in V100 GPU days, where * means our estimated time based on FLOPs, pretraining data, and the number of epochs for the methods that do not report their pretraining time. VINDLU uses ViT-B/16 while VINDLU-L uses ViT-L/16 as video encoders. For fair comparisons, we de-emphasize the CLIP-based methods since they use a lot more pretraining data than all other approaches. Our results indicate that VINDLU achieves competitive or even better than state-of-the-art results while also being simple and efficient.

| Method | #PT | SSv2-label | | SSv2-template | | Avg |
|---|---|---|---|---|---|---|
| | | R1 | R5 | R1 | R5 | |
| CLIP4Clip [40] | 400M | 43.1 | 71.4 | 77.0 | 96.6 | 77.9 |
| Singularity [25] | 17M | 47.4 | 75.9 | 77.6 | 96.0 | 80.0 |
| VINDLU | 5M | 51.2 | 78.8 | 82.2 | 98.9 | 82.7 |
| | 17M | 53.0 | 80.8 | **86.2** | 99.4 | **84.6** |
| | 25M | **53.1** | **81.8** | 83.3 | **100** | 84.4 |

Table 3. Comparison with state-of-the-art text-to-video retrieval methods on the temporally-heavy SSv2-Label [25] and SSv2-Template datasets [25]. #PT denotes the amount of pretraining data. Averaged numbers are the average of Recal@{1,5,10} on these two datasets. CLIP-based models are de-emphasized for fairer comparisons. We observe that VINDLU achieves the best performance, which demonstrates its ability to reason about complex temporal dependencies in the video data.

standing tasks. Our empirical insights enable us to develop a step-by-step recipe for effective video-language (VidL) pretraining, which leads to a highly performant VidL model, dubbed VINDLU. Compared to the existing VidL approaches, our method achieves competitive or even better results on 9 VidL benchmarks while also being simpler and more efficient. While our paper does not provide any novel individual contributions, we believe that our empirical insights and our VidL pretraining recipe will be useful and help advance further research in the VidL domain.

| Method | #PT | ANet | MSR-QA | MSR-MC | TVQA |
|---|---|---|---|---|---|
| ClipBERT [26] | 0.2M | - | 37.4 | 88.2 | - |
| ALPRO [28] | 5M | - | 42.1 | - | - |
| JustAsk [67] | 69M | 38.9 | 41.5 | - | - |
| VideoCLIP [64] | 136M | - | - | 92.1 | - |
| All-in-one [55] | 138M | - | 44.3 | 92.0 | - |
| MERLOT [75] | 180M | 41.4 | 43.1 | 90.9 | 78.7 |
| VIOLET [16] | 138M | - | 43.9 | 91.9 | - |
| Singularity [25] | 17M | 44.1 | 43.9 | 93.7 | - |
| OmniVL [57] | 17M | - | 44.1 | - | - |
| HERO [31] | 7.5M | - | - | - | 74.2 |
| FrozenBiLM [68] | 400M | 43.2 | 47.0 | - | 82.0 |
| VINDLU | 5M | 44.2 | 43.6 | 95.4 | **79.0** |
| | 17M | 44.6 | 43.8 | 93.8 | 78.8 |
| | 25M | **44.7** | **44.6** | **95.5** | **79.0** |

Table 4. Comparison with state-of-the-art video question-answering methods on ActivityNet-QA (ANet), MSRVTT-QA (MSR-QA), MSRVTT-MC (MSR-MC) and TVQA. #PT denotes the amount of pretraining data. We gray out FrozenBiLM [68] as it is much larger than our model (1.2B vs 207M parameters). VINDLU achieves competitive results across all four datasets.

**Acknowledgements.** We thank Y. Lin, M. Islam, A. Madasu and M. Gramopadhye for helpful discussions. This work was supported by the Sony Faculty Innovation award, Lilly Endowment Inc. via Indiana University Pervasive Technology Institute, Laboratory for Analytic Sciences via NC State University, and NSF-AI Institute DRL211263.

## 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. 1, 4

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021. 1, 2, 3, 4, 5, 6, 8

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. A clip-hitchhiker's guide to long video retrieval. *arXiv preprint arXiv:2205.08508*, 2022. 2, 3, 4, 8

[4] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 4

[5] 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. 2, 5

[6] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the" video" in video-language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2917–2927, 2022. 2, 3, 4, 5

[7] Jaeseok Byun, Taebaek Hwang, Jianlong Fu, and Taesup Moon. Grit-vlp: Grouped mini-batch sampling for efficient vision and language pre-training. In *European Conference on Computer Vision*, pages 395–412. Springer, 2022. 3

[8] 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. 1

[9] 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. 1

[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 *European conference on computer vision*, pages 104–120. Springer, 2020. 3

[11] 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. 4

[12] 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. 1, 2, 4

[13] Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, et al. An empirical study of training end-to-end vision-and-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18166–18176, 2022. 2, 3

[14] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020. 5

[15] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *arXiv preprint arXiv:2205.09113*, 2022. 6

[16] 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, 4, 5, 6, 8

[17] 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. 2, 4

[18] Zijian Gao, Jingyu Liu, Sheng Chen, Dedan Chang, Hao Zhang, and Jinwei Yuan. Clip2tv: An empirical study on transformer-based methods for video-text retrieval. *arXiv preprint arXiv:2111.05610*, 2021. 2, 4

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 6

[20] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17980–17989, 2022. 3

[21] Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 958–959, 2020. 4

[22] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021. 3

[23] 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. 1, 3, 4

[24] 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. 1, 2

[25] Jie Lei, Tamara L Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. *arXiv preprint arXiv:2206.03428*, 2022. 2, 3, 4, 5, 6, 8

[26] 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. 1, 2, 4, 8

[27] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018. 7

[28] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4953–4963, 2022. 4, 8

[29] 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. 3

[30] 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. 6

[31] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020. 1, 4, 8

[32] 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. 2, 4, 5, 7, 8

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[34] Yan-Bo Lin, Jie Lei, Mohit Bansal, and Gedas Bertasius. Eclipse: Efficient long-range video retrieval using sight and sound. *arXiv preprint arXiv:2204.02874*, 2022. 2, 8

[35] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*, 2019. 4

[36] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 3

[37] 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. 2, 6

[38] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 3

[39] 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, 4

[40] 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. 1, 2, 3, 4, 5, 8

[41] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. 1, 2

[42] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3163–3172, 2021. 5

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

[44] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning for action recognition. *arXiv preprint arXiv:2206.13559*, 2022. 5

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3, 4

[46] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 6

[47] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17959–17968, 2022. 2

[48] 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. 1, 6

[49] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, 2022. 3

[50] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019. 4

[51] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 3

[52] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learn-

ers for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022. 4, 6

[53] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 5

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[55] 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. 2, 4, 5, 7, 8

[56] Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. Reconstruction network for video captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7622–7631, 2018. 4

[57] 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. 2, 3, 4, 5, 6, 7, 8

[58] Jinpeng Wang, Yixiao Ge, Guanyu Cai, Rui Yan, Xudong Lin, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Object-aware video-language pre-training for retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3313–3322, 2022. 4

[59] Jianfeng Wang, Xiaowei Hu, Zhe Gan, Zhengyuan Yang, Xiyang Dai, Zicheng Liu, Yumao Lu, and Lijuan Wang. Ufo: A unified transformer for vision-language representation learning. *arXiv preprint arXiv:2111.10023*, 2021. 3

[60] Teng Wang, Wenhao Jiang, Zhichao Lu, Feng Zheng, Ran Cheng, Chengguo Yin, and Ping Luo. Vlmixer: Unpaired vision-language pre-training via cross-modal cutmix. In *International Conference on Machine Learning*, pages 22680–22690. PMLR, 2022. 3

[61] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. 3

[62] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018. 5

[63] 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. 4, 7

[64] 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. *arXiv preprint arXiv:2109.14084*, 2021. 1, 2, 8

[65] 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. 1, 3, 4

[66] 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. 2, 3, 4, 7, 8

[67] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1686–1697, 2021. 8

[68] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *arXiv preprint arXiv:2206.08155*, 2022. 7, 8

[69] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19163–19173, 2022. 3

[70] Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. Causal attention for vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9847–9857, 2021. 3

[71] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 3

[72] 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. 4, 7

[73] 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. 4, 7

[74] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 3

[75] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651, 2021. 1, 2, 5, 6, 8

[76] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *arXiv preprint arXiv:2111.08276*, 2021. 3

[77] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer.

Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022. 3

[78] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2021. 3

[79] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049, 2020. 3

[80] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755, 2020. 1, 6

[81] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 695–712, 2018. 4