Test of Time: Instilling Video-Language Models with a Sense of Time

Bagad, P.; Tapaswi, M.; Snoek, C.G.M.

DOI
10.48550/arXiv.2301.02074
10.1109/CVPR52729.2023.00247

Publication date
2023

Document Version
Author accepted manuscript

Published in
CVPR 2023

License
Unspecified

Citation for published version (APA):
Test of Time: Instilling Video-Language Models with a Sense of Time

Piyush Bagad
University of Amsterdam

Makarand Tapaswi
IIIT Hyderabad

Cees G.M. Snoek
University of Amsterdam

bpiyush.github.io/testoftime-website

Abstract

Modelling and understanding time remains a challenge in contemporary video understanding models. With language emerging as a key driver towards powerful generalization, it is imperative for foundational video-language models to have a sense of time. In this paper, we consider a specific aspect of temporal understanding: consistency of time order as elicited by before/after relations. We establish that seven existing video-language models struggle to understand even such simple temporal relations. We then question whether it is feasible to equip these foundational models with temporal awareness without re-training them from scratch. Towards this, we propose a temporal adaptation recipe on top of one such model, VideoCLIP, based on post-pretraining on a small amount of video-text data. We conduct a zero-shot evaluation of the adapted models on six datasets for three downstream tasks which require varying degrees of time awareness. We observe encouraging performance gains especially when the task needs higher time awareness. Our work serves as a first step towards probing and instilling a sense of time in existing video-language models without the need for data and compute-intensive training from scratch.

1. Introduction

Self-supervised pretraining at scale on multimodal web corpora tied with powerful architectures [107] has led to foundational models [12] for images [2, 49, 59, 83, 84] and videos [2, 6, 26, 109, 119, 126]. These models have enabled remarkable improvements on a plethora of downstream video-language tasks such as video-text retrieval, video question-answering, and action recognition. Given the cost and difficulty of video annotations, even for a small amount of downstream data, such foundational models are emerging as the de-facto backbone for zero-shot [119, 122, 127] and few-shot generalization [2]. However, it remains unclear if these video-language models capture essential properties of a video beyond what can be learned from static images, most notably: time.

Many before us have shown that existing video-language models [6, 57, 66, 119] can achieve impressive performance on several video benchmarks [22, 41, 120] without reliably encoding time [13, 56, 59]. For example, Buch et al. [13] show that a model that uses a single (carefully selected) frame often outperforms recent video-language models [57, 119] on standard video benchmarks such as MSR-VTT [120]. Lei et al. [56] report similar findings with a single-frame pretraining approach. These findings hint at a lack of time awareness in video models. However, it remains unclear if these findings are caused, indeed, by the lack of time in video models or whether the benchmarks themselves do not mandate time awareness. Furthermore, there is no clear definition of what it means for a model to be time aware. In this paper, we strive to shed light on all these factors of time awareness in video-language models.

As a first step, we consider a simple notion of understanding time, i.e., understanding temporal relations such as before and after [4]. Consider the task presented in Fig. 1. A time invariant model shall be able to associate (A) with (1)
or (2) and (B) with (3) or (4) based on static frames alone. But to distinguish between (1) and (2), one needs to be able to understand time order and connect it across video and language\footnote{Answers: (A)-(2), (B)-(1), (C)-(4), (D)-(3).}. Thus, the first question we ask in Section 3: do the representations learnt by foundational video-language models encode this sense of time? To reliably attribute lack of time awareness to models and not existing benchmarks, we design our own synthetic dataset to probe models for this sense of time. We create video-language pairs that show a sequence of two events. Then, we alter the order of events either in the text or the video and check if models can connect the order in video and language. We find that existing video-language models indeed struggle to associate the time order across video and language.

In light of these findings, the second question we ask in Section 4 is: can we adapt a video-language model, without expensive re-training from scratch, to instill this sense of time? Towards this, we take inspiration from literature on understanding time in natural language, where there has been much work on developing time aware language models \cite{20,36,37,130,131}. Our objective is to instill time awareness in a video-language model without having to pre-train from scratch. To do that, we propose TACT: Temporal Adaptation by Consistent Time-ordering based on two key components: (i) we artificially create samples that provide temporal signal, for example, by flipping the order of events in the video or the text, and (ii) we introduce a modified contrastive loss to learn time order consistency based on these samples. Instead of training from scratch, we adapt an existing video-language model, VideoCLIP \cite{61}, using the paradigm of post-pretraining on a small amount of video-text data \cite{66,121}. We demonstrate the effectiveness of TACT in connecting the time order in video and language on four diverse real datasets in Section 5.

Finally, in line with the original motive of video-language models for zero-shot generalization, we evaluate in Section 6 our TACT-adapted model for three sets of tasks on six downstream datasets which require a varying degree of time awareness. On tasks that need higher time awareness, with the appropriate choice of adaptation dataset, TACT outperforms a strong baseline that is based on post-pretraining on canonical clip-text pairs without consideration of time-order.

In summary, our contributions are: (i) We show that existing video-language models struggle to associate time order in video and language through controlled experiments on synthetic data and several evaluations on real datasets. (ii) Based on VideoCLIP \cite{119}, we propose TACT, a method for temporal adaptation using this time order consistency without having to pretrain from scratch. (iii) We demonstrate improved zero-shot generalizability of TACT-adapted models on tasks that require higher time awareness.

2. Background and Related Work

We briefly discuss recent advances in video-language models followed by their consideration of time.

Foundational video-language models. Large-scale datasets, self-supervision, and the advent of Transformers \cite{107} have led to the emergence of powerful encoders for images \cite{21,39,103}, videos \cite{5,11,24,104,117}, language models \cite{19,64,69,86} and even universal encoders \cite{32,46}. These encoders form the basis for several vision-language foundational models. Popular image-language models such as CLIP \cite{83} and ALIGN \cite{49} are trained on massive datasets by using web images and alt-text. Similarly, video-language models are catching up and can be categorised into two broad directions: (i) adapting image-language models for videos \cite{8,23,50,51,63,66,71,110,112,121}, and (ii) pure video-based models that are learned using large video-text datasets \cite{3,7,25,31,58,62,67,68,95}. Recently, a new paradigm of post-pretraining has emerged where an existing image- or video-language model goes through another stage of self-supervised pretraining on a small amount of video data before it is evaluated on downstream tasks \cite{66,121}. This is promising as it circumvents the prohibitive cost of pretraining on large datasets from scratch. In \cite{66}, the post-pretraining uses time-invariant mean-pooling, while \cite{121} strives to bridge the domain gap between image captions and video subtitles. In contrast, our proposed temporal adaptation involves post-pretraining of VideoCLIP \cite{119} with a small amount of data that instills the model to learn the time-order of events in a video.

Time in vision. Time separates videos from static images or an unordered set of frames. While modeling time remains a challenge, it also presents a natural source of supervision that has been exploited for self-supervised learning. For example, as a proxy signal by posing pretext tasks involving spatio-temporal jigsaw \cite{1,44,53}, video speed \cite{10,17,48,91,124}, arrow of time \cite{78,80,114}, frame/clip ordering \cite{25,70,90,97,118}, video continuity \cite{61}, or tracking \cite{45,108,113}. Several works have also used contrastive learning to obtain spatio-temporal representations by (i) contrasting temporally augmented versions of a clip \cite{47,77,81}, or (ii) encouraging consistency between local and global temporal contexts \cite{9,18,85,123}. Nevertheless, it remains unclear if the learnt representations actually encode time reliably. Time-aware features have also been explored for specific downstream tasks such as action recognition \cite{30,100,101}. There has been some very recent work on evaluating self-supervised video representations \cite{87,99} on their temporal recognition ability instead of only relying on time as a guidance for training.

In the same spirit, a related direction pursues evaluation and benchmarking of time awareness in video datasets \cite{88}, models \cite{13,14,30,56,89,125} or both \cite{43,92}. Huang et
al. [43] measure the effect of motion on temporal action recognition to find that only a subset of classes in UCF-101 and Kinetics-400 require motion information. Ghodrati et al. [30] propose new tasks to evaluate temporal asymmetry, continuity and causality in video models. Our work derives inspiration from these but applies more generally to video-language models as language provides a basis for open-world generalization.

**Time in language.** Time has also been extensively studied in the natural language literature. Early works identified temporal structures in language such as temporal prepositions and quantifiers [4, 79]. More recent literature focuses on tasks such as extracting temporal relations [35, 72–74], as well as temporal reasoning [36, 37, 82, 130, 131]. For example, Han et al. [36, 37] and Zhou et al. [131] pretrain language models specifically to focus on understanding temporal relations such as before, after, during, etc. The emergence of large language models has also spurred an increased interest in developing benchmarks to test for time awareness in these models [20, 75, 76, 102, 106, 129]. For example, Ning et al. [75] propose a new benchmark of reading comprehension with questions involving before/after relations. Since temporal relations in language are grounded in the video, we draw inspiration from [36, 37, 131] and aim to instill time awareness in video-language models.

**Time in video-language models** appears implicitly in tasks like video-text alignment [38, 98] and temporal grounding [42, 60]. In this work, we focus on self-supervised video-language models that can generalize to a variety of tasks rather than models designed for a specific task, e.g., temporal grounding. Some recent works have shown the under-utilisation of time in classic video-text benchmarks such as MSR-VTT [120], YouCook [132], ActivityNet [22], and DiDeMo [41]. For example, Han et al. [36, 37] and Zhou et al. [131] discover that on several benchmarks, using only one or few frames or clips achieves competitive performance. Adaptations of the popular CLIP architecture for videos (e.g., CLIP4Clip [66]) show that weighted pooling of frames already achieves impressive performance on retrieval benchmarks.

These raise some key questions: do existing video-language models truly understand time in the sense of correctly associating order of events in language and video? If not, can we adapt them to instill time awareness? Our work addresses these questions. There has been some work in using time-order across video and language as a source of self-supervision. Specifically, concurrent to our work, both Sun et al. [96] and Cao et al. [15] propose fine-grained temporal alignment between video and text as the pretraining objective. Different from these works, we consider the notion of time-order and we aim to adapt a given video-language model using post-pretraining, which circumvents the need for a new round of compute-intense pretraining.

![Figure 2](image.png)

**3. Do Video-Language Models Sense Time?**

Probing video-language models for temporal understanding is an open direction of research. In this work, we consider a specific sense of temporal understanding: consistency in the order of events in a video with the associated textual description. For example, consider a text description: A red circle appears before a yellow circle. This imposes an order constraint on the video stream to have the event red circle appears happen before the event yellow circle appears. Can existing video-language models connect time-order in text with that in video? To answer this, we design an experiment with synthetic data.

**Synthetic dataset.** We construct simple videos that comprise a pair of events such as the ones mentioned above. We generate $N=180$ video-language pairs as a combination of $C=6$ colors, $S=3$ shapes, and $|\tau|=2$ temporal relations: before and after. The corresponding caption describes the order of events connected with a before/after temporal relation. We call this caption as an attractor since it is consistent with the time-ordering in the video. Likewise, we construct a distractor where we flip the order of event descriptions while retaining the temporal relation. An example pair is illustrated in Fig. 2 (left). Ideally, a time aware video-language model should be able to associate the video with the temporally consistent text, or vice versa. We refer to this task as time-order consistency check. To rule out the possibility that synthetic videos are out-of-distribution, we also perform the same experiment with canonical clips with a single event and the text describes that same event as shown in Fig. 2 (right). We refer to this as the control task.

**Choice of models.** We consider recent video-language models, broadly categorized into three groups: (i) image-language models like CLIP [83] that are adapted to videos [23,66,128], (ii) pure video-language models trained on a contrastive learning objective [6,16,119], and (iii) pure video-language models trained on a masking objective [29].

**Findings.** We evaluate video-to-text and text-to-video retrieval on both time-order consistency and control tasks.
From Tab. 1, we observe that while most video-language models perform well on the control task, all of them struggle and perform on par with random chance on the temporal task. This gap in performance deserves attention given the importance of time in videos. Note that while synthetic data allows for controlled experiments, we also expand this evaluation to real video datasets in the following section.

4. Adaptation by Consistent Time-Ordering

We describe a post-pretraining recipe to instill a video-language model with a sense of time. We propose TACT: Temporal Adaptation by Consistency of Time-order, that is based on two key components: (i) we artificially create samples that provide temporal signals, e.g., by flipping the order of events; and (ii) we introduce a modified contrastive loss to learn temporal consistency based on these samples. We start by defining the notation and then describe the key components of our adaptation recipe.

**Preliminaries.** Let $\mathcal{V}$ be the space of video clips and $\mathcal{T}$ be the space of text clips. Consider two non-overlapping video clips $v_i, v_j \in \mathcal{V}$. Let $\zeta_i, \zeta_j \in \mathcal{T}$ be their respective captions. Let $\tau$ be a temporal relation, $\tau \in \{\text{before}, \text{after}\}$. Then, we denote a \textit{stitched} and time-order consistent clip as $(u_{ij}, t_{ij})$, where $u_{ij} := [v_i; v_j], t_{ij} := [\zeta_i; \zeta_j]$, and $[\cdot; \cdot]$ denotes concatenation. Note that depending on $\tau$, the order of $v_i$ and $v_j$ may need to change in $u_{ij}$. For brevity, we drop the subscripts and refer to the stitched pair as $(u, t)$ unless stated otherwise.

**Time-order reversal.** The classic contrastive learning paradigm for video-language models aligns components of a video clip $v_i$ with its text counterpart $\zeta_i$ and contrasts against other texts $\zeta_j$ that usually describe a completely different clip. This makes such models ignore the finer details of temporal understanding as it is easier to contrast the negatives by simply focusing on the objects or the scene. This is evident from simple bag-of-word-like methods that are shown to work well for contrastive learning, both on the visual (e.g., CLIP4Clip [66]) and textual (e.g., MILNCE [67]) modalities. We hypothesize that unless there are negatives in a contrastive setup that contain the same scenes and objects, models need not learn a sense of time. Thus, we present a simple strategy to generate negatives that force the learning process to focus on the temporal order.

We define a time-order reversal function $\mathcal{T}$ that operates on the stitched video clip or text description and temporally swaps its constituents:

$$\mathcal{T}(u) = \mathcal{T}([v_i; v_j]):=[v_j; v_i], \quad \text{and}$$  
$$\mathcal{T}(t) = \mathcal{T}([\zeta_i; \zeta_j]):=[\zeta_j; \zeta_i].$$

An illustration of $\mathcal{T}$ is shown in Fig. 3. Note that $\mathcal{T}$ does not reverse the actual video, i.e., time does not flow backwards, but only changes the order in which events happen in the stitched clips. Our objective is to train a model that is able to distinguish between the original pair $(u, t)$ and time-reversed versions $(u, \mathcal{T}(t))$, and $(\mathcal{T}(u), t)$.

**Loss function.** We assume access to an existing pre-trained video-language model with a visual encoder $f_v$ and text encoder $g_\phi$. We obtain the video encoding $z_u := f_v(u) \in \mathbb{R}^d$ and the text encoding $z_t := g_\phi(t) \in \mathbb{R}^d$. Our goal is to adapt $\Theta = \{\theta, \phi\}$ via post-pretraining such that the resulting model is time aware while maintaining its original performance on tasks such as retrieval. As we aim to use a small amount of data, we only update some parameters of
the model (Θ), such as the last few layers.

We now introduce our recipe for temporal adaptation based on the InfoNCE loss [105] to learn time-order sensitive video-text correspondence. For a positive (or time-order consistent) video-text pair (u, t), we first define a forward loss where the stitched pair is in its original time-order.

\[ L_f = \sum_{(u,t) \in B} (\text{TNCE}(z_u, z_t) + \text{TNCE}(z_t, z_u)) , \tag{3} \]

where TNCE is the Noise Contrastive Estimation (NCE) loss for temporal adaptation, defined as:

\[ \text{TNCE}(z_u, z_t) := -\log \frac{\exp(z_u \cdot z_t)}{\sum_{t' \in B_t} \exp(z_u \cdot z_{t'})} + \alpha_{\text{time}} , \tag{4} \]

where \( B \) is the batch of (u, t) pairs and \( B_t \) specifically refers to other stitched text captions in the batch. \( \alpha_{\text{time}} \) accumulates negatives defined using time-order reversal as:

\[ \alpha_{\text{time}} = \alpha_{\text{same}} \exp(z_u \cdot z_{T(t)}) + \alpha_{\text{cross}} \sum_{t' \in B_t \setminus \{t\}} \exp(z_u \cdot z_{T(t')}) , \tag{5} \]

where \( \alpha_{\text{same}} \) controls the effect of contrasting text from the same sample but with reversed text time-order, i.e., \( T(t) \), and \( \alpha_{\text{cross}} \) encourages the model to contrast between reversed versions of other text captions, i.e., \( T(t') \). Note that when both \( \alpha_{\text{same}} \) and \( \alpha_{\text{cross}} \) are 0, we revert back to the standard NCE formulation, albeit on stitched pairs. While Eq. (4) corresponds to the video-text loss \( \text{TNCE}(z_u, z_t) \), the text-video loss \( \text{TNCE}(z_t, z_u) \) is defined symmetrically. Furthermore, we also apply a reverse loss \( L_r \) to bring time-order reversed versions of both the video and the text together. Note that as we consider (u, t) as a positive pair, \( (T(u), T(t)) \) also form a positive pair,

\[ L_r = \sum_{(T(u), T(t)) \in B} (\text{TNCE}(z_{T(u)}, z_{T(t)}) + \text{TNCE}(z_{T(t)}, z_{T(u)})) . \tag{6} \]

Here, the TNCE term operates on time-reversed clips and \( \alpha_{\text{time}} \) contrasts \( (T(u), T(t)) \) with un-reversed text clips in the batch \( (T(u), t) \). The overall loss function is defined as:

\[ L = L_f + \beta L_r . \tag{7} \]

Depending on the choice of loss coefficients \( \alpha_{\text{same}}, \alpha_{\text{cross}}, \beta \in \{0, 1\} \), we can vary properties of the adapted model. For example, setting \( \alpha_{\text{same}} = 1 \) encourages high sensitivity to time-order reversal. As we will see empirically, the loss coefficients also provide the flexibility to adapt the model to various datasets.

We illustrate this temporal extension of the contrastive loss in Fig. 3 (best seen in colour). \( T \) illustrates the time-order reversal function. The top half corresponds to \( L_f \) while the bottom half visualizes \( L_r \). In particular, the top-left quadrant alone corresponds to the standard contrastive loss on stitched pairs. While the green diagonal terms are positive pairs, the red diagonal terms are the strongest drivers for instilling temporal understanding in the model.

### Table 2. Statistics of datasets we consider for temporal adaptation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
<th>Ego Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMPO</td>
<td>3,904</td>
<td>28,427</td>
<td>411</td>
<td>1,000</td>
</tr>
<tr>
<td>ActivityNet</td>
<td>7,440</td>
<td>95,474</td>
<td>453</td>
<td>906</td>
</tr>
<tr>
<td>Charades</td>
<td>5,262</td>
<td>99,928</td>
<td>500</td>
<td>1,000</td>
</tr>
<tr>
<td>Charades-Ego</td>
<td>2,679</td>
<td>155,306</td>
<td>500</td>
<td>1,000</td>
</tr>
</tbody>
</table>

5. Experiments: TACT Ablations

**Base model.** We demonstrate the effectiveness of TACT as an adaptation recipe on top of VideoCLIP [119] owing to its simple architecture, contrastive objective, and use pre-computed S3D [116] features that enable faster experimentation and allow encoding a long temporal context (~32 secs). We initialize Θ from the model pretrained on HowTo100M [68] and post-pretrain on multiple datasets.

**Datasets.** One of our key objectives is to post-pretrain on a small amount of data in contrast to massive pretraining datasets such as WebViD2M [7] or HowTo100M [68]. We consider dense video captioning datasets that offer sufficient diversity in terms of size, backgrounds, clip durations, viewpoints, and activities. Specifically, we experiment with: (i) **TEMPO** [40]: the subset of stitched diverse third-person videos from DiDeMo [41] with text descriptions for fixed 5s segments that contain before/after relations; (ii) **ActivityNet Captions** [55]: a dense video captioning dataset with human-centric actions; (iii) **Charades** [93]: a scripted indoor daily human activities video dataset; and (iv) **Charades-Ego** [91]: similar to Charades, scripted human activities from the egocentric viewpoint. To construct stitched clips, we randomly sample any two non-overlapping clip-text pairs in the video. Since we require stitched clips instead of raw videos, we create new splits for each dataset (see Tab. 2).

**Evaluation metrics.** We evaluate the post-pretrained model using two sets of metrics: (i) standard retrieval metrics, recall \( R@1, R@5, R@10 \) and median-rank evaluated on stitched video-text clips; and (ii) time-order consistency, i.e., the fraction of videos for which the model correctly associates text that is time order consistent with the video:

\[ A_{\text{time}} := \frac{1}{|\mathcal{D}|} \sum_{(u,t) \in \mathcal{D}} \mathbb{1}[d(z_u, z_t) < d(z_u, z_{T(t)})] , \tag{8} \]

where \( (u, t) \) are time-order consistent pairs, \( \mathcal{D} \) is the dataset, and \( d(\cdot, \cdot) \) is a distance metric based on cosine similarity.

**Post-pretraining details.** We freeze the word embeddings
and layers 1 to 5 for both the video and text encoders in VideoCLIP. For adaptation, we use the Adam optimizer [54] with learning rate $5e^{-6}$, batch size 32 trained on a single node with 4 GeForce GTX 1080 GPUs. On TEMPO, we train for 60 epochs while on the other datasets, we train for 10 epochs and pick the checkpoint that maximizes the geometric mean of $R@1$ and $A_{\text{time}}$ on the respective validation set. A typical adaptation run takes about 1-3 hours.

**Evaluation on the test set.** Results in Tab. 3 show that TACT* with optimal loss coefficients outperforms TACT† (all 0 loss coefficients) and the zero-shot baseline (no post-pretraining similar to the synthetic data experiment), both on the retrieval and time-order consistency tasks. This indicates the robustness of the adaptation.

**Impact of loss coefficients.** Choosing appropriate values for loss coefficients $\Theta := \{\alpha_{\text{same}}, \alpha_{\text{cross}}, \beta\}$ allows the model to learn various aspects and adapt using different datasets. On each dataset, we vary $\Theta \in \{0, 1\}^3$ and find the best configuration based on the GeometricMean($R@1, \max(A_{\text{time}} - 50, 0)$) on the validation sets. The above metric ensures the geometric mean is not overpowered by $R@1$. The results are shown in Tab. 4. As $\alpha_{\text{same}}$ is directly responsible for discriminating between original and time-reversed orders, unsurprisingly, setting it to 1 is necessary to achieve the best results on $A_{\text{time}}$ for all the datasets. For TEMPO and Charades-Ego, using all loss components (all 1) provides the best results, whereas $\alpha_{\text{cross}}=1$ and $\beta=0$ achieves a better trade-off for ActivityNet and Charades. Choosing $\beta=1$ leads to an improve-ment in retrieval performance for TEMPO and Charades-Ego but leads to a decline for ActivityNet and Charades. We attribute this to the number of unique videos in the train set for these datasets. As ActivityNet and Charades have more videos than TEMPO or Charades-Ego (see train $N_\ell$ Tab. 2) additional positives introduced by setting $\beta=1$ are not necessary and in fact, hurt performance. Finally, we note that carefully setting the value of $\alpha_{\text{same}}$ provides a convenient trade-off between spatial and temporal understanding. Please refer to the supplement for detailed experiments.

**What makes temporal adaptation hard?** We observe a large gap in $A_{\text{time}}$ between TEMPO and ActivityNet. We hypothesize that the distance (in seconds) between the two clips ($\Delta_{\text{time}}$) in a stitched clip is strongly correlated with the difficulty of adaptation. Intuitively, it is easier to infer the time-order consistency for a stitched clip $u$ with text $t$ that has distant constituent clips $v_i, v_j$ since the objects and scene could be vastly different. In contrast, it is harder to discern the correct time-order when the constituent clips are closer in time. Fig. 4 shows distribution of $\Delta_{\text{time}}$ for each dataset. Indeed, $\Delta_{\text{time}}$ in ActivityNet (58.8s) is much higher than that in TEMPO (6.4s) making the task harder on TEMPO. To further test our hypothesis, we conduct a controlled experiment where we gradually vary the distribution of $\Delta_{\text{time}}$ for Charades-Ego to match it to that of TEMPO. We find a strong correlation ($\rho=0.92$) between $\Delta_{\text{time}}$ and hardness of adaptation. More details are in the supplement.

### Table 3: Results for the best TACT model on test sets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Retrieval</th>
<th>Time-order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$R@1$</td>
<td>MedR $\downarrow$</td>
<td>$A_{\text{time}}$ $\uparrow$</td>
</tr>
<tr>
<td>TEMPO</td>
<td>Zero-shot</td>
<td>3.7</td>
<td>49.0</td>
</tr>
<tr>
<td></td>
<td>TACT†</td>
<td>7.7</td>
<td>13.0</td>
</tr>
<tr>
<td></td>
<td>TACT*</td>
<td>9.3</td>
<td>9.0</td>
</tr>
<tr>
<td>ActivityNet</td>
<td>Zero-shot</td>
<td>1.1</td>
<td>44.0</td>
</tr>
<tr>
<td></td>
<td>TACT†</td>
<td>5.8</td>
<td>34.0</td>
</tr>
<tr>
<td></td>
<td>TACT*</td>
<td>5.8</td>
<td>35.0</td>
</tr>
<tr>
<td>Charades</td>
<td>Zero-shot</td>
<td>1.3</td>
<td>170.0</td>
</tr>
<tr>
<td></td>
<td>TACT†</td>
<td>5.3</td>
<td>38.5</td>
</tr>
<tr>
<td></td>
<td>TACT*</td>
<td>5.7</td>
<td>35.0</td>
</tr>
<tr>
<td>Charades-Ego</td>
<td>Zero-shot</td>
<td>1.6</td>
<td>64.0</td>
</tr>
<tr>
<td></td>
<td>TACT†</td>
<td>6.4</td>
<td>35.0</td>
</tr>
<tr>
<td></td>
<td>TACT*</td>
<td>10.1</td>
<td>28.5</td>
</tr>
</tbody>
</table>

Figure 4. Time-distance between stitched clips in datasets for temporal adaptation ($\Delta_{\text{time}}$). TEMPO has stitched clips close to each other while those in Charades-Ego are farthest apart suggesting a correlation between $\Delta_{\text{time}}$ and difficulty of temporal adaptation.

6. **Experiments: Downstream Evaluation**

The goal of video-language foundation models is to generalize in a zero- or few-shot manner to a diverse range of downstream tasks. We evaluate TACT models on three sets of downstream tasks that need low-to-high time awareness.

**Baseline: Standard post-pretraining.** Comparing our temporally adapted models with pretrained VideoCLIP is not fair since adapted models see data beyond the pretraining phase. In addition to the zero-shot comparison, we compare against a baseline model that is trained for standard video-text retrieval on the same datasets as temporal adaptation. Instead of using stitched clips, we use simple canonical pairs, i.e., $(v_i, \zeta_i)$ instead of $(u_{ij}, t_{ij})$.  

![Figure 4](image-url)
Evaluating TACT adapted models on synthetic data.

On the video-to-text variant, TACT adapted on TEMPO achieves 64.4%, ActivityNet 52.5%, Charades 65.0%, Charades-Ego 85.6%. This is usually higher than the performance that non-adapted models achieve in Tab. 1. This highlights that TACT models learn useful signals to match time-order in video and language.

I. Text-to-video retrieval. We consider two widely used benchmarks: MSR-VTT [120] and YouCookII [132] and adopt standard retrieval metrics. Recent work has identified a bias for spatial understanding in these datasets [8,13,43,56,59,66]. Thus, we consider this class of tasks as requiring low time awareness. As shown in Tab. 5 set I, on MSR-VTT [120], we observe that TACT is worse (marked in red) or at par with the baseline across datasets. This aligns well with findings in [13,56] that these benchmarks do not need time awareness. On YouCookII [132], TACT models based on Charades(-Ego) outperform the baseline (marked in green). We believe this is a consequence of a lower domain shift between YouCookII and Charades.

II. Temporal video QA. Next, we use subsets of recently released multiple-choice video question answering benchmarks: Next-QA [115] and AGQA [34]. The idea is to check if we can probe models for temporal understanding by asking questions with temporal language. Buch et al. [13] identify a subset of Next-QA, dubbed as ATP-hard, with a higher concentration of temporally challenging data. For AGQA, we pick a subset of ~6k questions that explicitly have a question with before/after relation. We consider

\[ \text{Accuracy, mAP} \]

Available here: github.com/StanfordVL/atp-video-language

Table 4. Impact of loss coefficients for TACT post-pretraining on validation sets of various datasets. For each dataset, the corresponding color-marked row denotes the best configuration based on the geometric mean of \( R@1 \) and \( A_{\text{time}} \). TACT is able to connect time-order in video and language while maintaining its retrieval capabilities across several datasets.

<table>
<thead>
<tr>
<th>Loss coefficients</th>
<th>TEMPO</th>
<th>ActivityNet</th>
<th>Charades</th>
<th>Charades-Ego</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_{\text{time}} )</td>
<td>( \alpha_{\text{cross}} )</td>
<td>( \beta )</td>
<td>( R@1 \uparrow )</td>
<td>( \text{MedR} )</td>
</tr>
<tr>
<td>Chance</td>
<td>0.1</td>
<td>500.0</td>
<td>50.0</td>
<td>0.1</td>
</tr>
<tr>
<td>None</td>
<td>10.6</td>
<td>23.4</td>
<td>29.9</td>
<td>18.2</td>
</tr>
<tr>
<td>TEMPO Baseline</td>
<td>12.0</td>
<td>29.3</td>
<td>37.3</td>
<td>21.5</td>
</tr>
<tr>
<td>TACT</td>
<td>12.8</td>
<td>26.5</td>
<td>35.7</td>
<td>20.4</td>
</tr>
<tr>
<td>ActivityNet Baseline</td>
<td>15.7</td>
<td>34.4</td>
<td>44.9</td>
<td>15.6</td>
</tr>
<tr>
<td>TACT</td>
<td>13.8</td>
<td>29.6</td>
<td>39.6</td>
<td>16.0</td>
</tr>
<tr>
<td>Charades Baseline</td>
<td>12.3</td>
<td>25.8</td>
<td>33.6</td>
<td>21.5</td>
</tr>
<tr>
<td>TACT</td>
<td>11.7</td>
<td>25.2</td>
<td>32.4</td>
<td>22.4</td>
</tr>
<tr>
<td>Charades-Ego Baseline</td>
<td>13.1</td>
<td>27.5</td>
<td>34.5</td>
<td>19.4</td>
</tr>
<tr>
<td>TACT</td>
<td>11.1</td>
<td>24.6</td>
<td>30.6</td>
<td>21.9</td>
</tr>
</tbody>
</table>

Table 5. Results on downstream zero-shot evaluation with tasks requiring increasing time awareness from I to III. None corresponds to direct evaluation of the VideoCLIP model on the downstream dataset. Green denotes an improvement for the TACT adapted model w.r.t. the baseline, red denotes a deterioration. As we move from tasks that need low to high time awareness, the effectiveness of TACT increases. See Sec. 6 for a more detailed discussion. The table is best viewed on screen in colour.
these benchmarks as requiring a moderate-high level of time awareness and AGQA in particular is also close to our adaptation task. We use accuracy as the standard metric. We observe (see Tab. 5 set II) that indeed TACT almost always outperforms (marked in green) baselines on both Next-QA and AGQA. TEMPO-adapted TACT seems to generalize particularly well on both benchmarks. Likewise, Charades-adapted TACT does well on AGQA since AGQA is also based on the Charades videos accounting for reduced domain shift. We affirm that temporal adaptation is useful, especially when the downstream tasks require it.

III. Action-to-video retrieval. Finally, we consider action recognition benchmarks such as Something-Something (SSv2) [33] and Temporal [88]. SSv2 was designed to capture richer temporal information [33, 56]. We follow Lei et al. [56], who propose the template-retrieval task that encourages temporal modelling and use their evaluation split using C=174 actions and K=12 videos per class. Interestingly, different actions in SSv2 require differing levels of time awareness. We create a subset SSv2 (events) with C-events=49 actions that have at least two verbs in the label as the occurrence of multiple verbs is indicative of multiple events occurring in sequence. Finally, we also evaluate against the Temporal benchmark [88], a combination of 50 action classes from SSv2 [33] and Kinetics-400 [52] for which temporal information is deemed to be essential for recognition. Similar to text-to-video retrieval, we use the action class as a text query and obtain a ranking over all videos. Different from the retrieval setup, since a single query has multiple correct answers (up to K=12 videos), we report mAP as the metric for these benchmarks. This task set needs high time awareness. Furthermore, unlike QA tasks in II, there is a shift in several (uncontrolled) factors as we move from temporal adaptation task to these tasks. From Tab. 5, we observe that TEMPO- and Charades-adapted models generalize well across set III benchmarks. ActivityNet-adapted TACT underperforms on SSv2 but outperforms the baseline on strongly temporal actions in SSv2 (events) and Temporal. Finally, TACT adapted on Charades-Ego is at-par or slightly worse than the baseline on SSv2 variants, and also on Temporal, perhaps due to the shift from egocentric to third-person videos. Overall, despite SSv2 and Temporal requiring high time awareness, TACT models show promising zero-shot generalization with the right choice of the adaptation dataset.

7. Discussion and Conclusion

Generalization to other temporal prompts. The time-order of events in language can be described using various sentence structures. Although we train video-language models using before/after relations, it is natural to ask if the model still correctly associates time-order for a different prompt such as First, . . . , then . . . . To systematically test this, we gather event pairs E1, E2 (E1 occurs before E2 in the video) for each sample in the validation set and stitch them using three prompts as follows: (i) E1 before E2, (ii) E2 after E1, (iii) First E1, then E2. As shown in Fig. 5, TACT-adapted models generalize well to a new prompt (iii). This substantiates the learning of time-order of events rather than merely learning the order of words in the sentence.

Limitations. While we present a promising way of instilling time in video-language models, our work is limited to the VideoCLIP [119] pretrained model. Our initial experiments with Frozen in Time [6] were not as promising, perhaps because it uses a much shorter temporal context (4 frames). Please see the supplement for results on more pretrained models. Furthermore, we consider a specific definition of time awareness derived from temporal relations like before/after. It is natural to ask if this can be extended to more general notions of temporality, e.g., as defined by Allen [4]. Finally, there can always be more downstream tasks considered such as (spatio-)temporal localization.

Conclusion. Given the essence of time in video-language models, we present a simple experiment based on synthetic data to test for time awareness. We find that existing models lack a sense of time defined in terms of consistency of order of events in video and language. To fill this gap, building upon VideoCLIP [119], we present TACT, a recipe to instill this sense of time in video-language models. Finally, we analyze the zero-shot generalizability of TACT-adapted models to a diverse set of tasks. We hope that this work provokes further probing and instilling time awareness in video-language models, and also inspires other adaptations of foundational models to solve various challenging tasks.

Acknowledgements. We acknowledge support from EL-LIS Amsterdam and the AMS Scholarship to Piyush. We thank Dr Dennis Koelma for their help with compute infrastructure and Dr Hazel Doughty for useful discussions.

Figure 5. Models trained by TACT with before/after relations generalize to a new kind of prompt such as First, . . . , then . . . indicating learning of the underlying true time-order of events.
References


[61] Qiang Ning, Zhili Feng, and Dan Roth. A structured learning approach to temporal relation extraction. In Empirical Methods in Natural Language Processing (EMNLP), 2017. 3


[63] Qiang Ning, Sanjay Subramaniam, and Dan Roth. An improved neural baseline for temporal relation extraction. In Empirical Methods in Natural Language Processing (EMNLP), 2019. 3


[67] Lyndsey C. Pickup, Zheng Pan, Donglai Wei, YiChang Shih, Changshui Zhang, Andrew Zisserman, Bernhard


[117] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin P. Murphy. Rethinking spatiotemporal feature learning for video understanding. *ArXiv*, abs/1712.04851, 2017. 2


[120] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 3, 7


