ProTéGé: Untrimmed Pretraining for Video Temporal Grounding by Video Temporal Grounding

Lan Wang⋆‡ Gaurav Mittal⋆† Sandra Sajeev† Ye Yu† Matthew Hall†
Vishnu Naresh Boddeti‡ Mei Chen†
†Microsoft ‡Michigan State University
{gaurav.mittal, yu.ye, mathall, ssajeev, mei.chen}@microsoft.com
{wanglan3,vishnu}@msu.edu

Abstract

Video temporal grounding (VTG) is the task of localizing a given natural language text query in an arbitrarily long untrimmed video. While the task involves untrimmed videos, all existing VTG methods leverage features from video backbones pretrained on trimmed videos. This is largely due to the lack of large-scale well-annotated VTG dataset to perform pretraining. As a result, the pretrained features lack a notion of temporal boundaries leading to the video-text alignment being less distinguishable between correct and incorrect locations. We present ProTéGé as the first method to perform VTG-based untrimmed pretraining to bridge the gap between trimmed pretrained backbones and downstream VTG tasks. ProTéGé reconfigures the HowTo100M dataset, with noisily correlated video-text pairs, into a VTG dataset and introduces a novel Video-Text Similarity-based Grounding Module and a pretraining objective to make pretraining robust to noise in HowTo100M. Extensive experiments on multiple datasets across downstream tasks with all variations of supervision validate that pretrained features from ProTéGé can significantly outperform features from trimmed pretrained backbones on VTG.

1. Introduction

Video temporal grounding (VTG) is the video-language multimodal task of localizing which part of an arbitrarily long untrimmed video can be best associated with a given natural language text query. VTG has a wide range of applications, such as information retrieval and robotics. Figure 1 shows a sample video-text pair for the VTG task and illustrates the primary challenge in grounding an unconstrained natural language text query in a long untrimmed video, namely, the need for a fine-grained understanding of the spatio-temporal dynamics in the video.

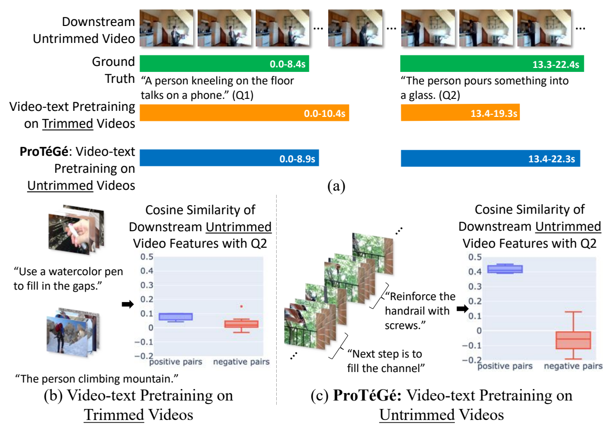

Figure 1. **(a) Comparison between video-text trimmed and untrimmed pretraining on grounding text Q1 and Q2 in an untrimmed video.** Untrimmed video-text pretraining shows stronger grounding capability. (b) and (c) show box plots of cosine similarity after joint video-text pretraining on trimmed and untrimmed videos, respectively between video features aligning with text (blue) and not aligning with text (red). We observe that compared to using trimmed videos (b), cosine similarities of video features aligning with text are higher and farther apart from that of video features not aligning with text when using untrimmed videos (c), thus illustrating the impact of untrimmed pretraining.

While there are multiple approaches for VTG, all existing methods, to the best of our knowledge, rely on video backbones pretrained on trimmed videos (such as Kinetics [15]) to obtain the visual features as part of their respective approaches. Such a design choice introduces a disconnect between the downstream VTG task on untrimmed videos and the trimmed videos used for pretraining the model from which video features are derived. For example, Fig 1 shows that the grounding predictions (in orange), when using a backbone jointly pretrained on trimmed videos and text, do not match adequately with the ground truth. Due to pretraining on trimmed videos, the video backbone is insensitive to temporal boundaries since the training objective is to associate an entire trimmed video to a label/text

⋆Authors with equal contribution.
This work was done as Lan Wang's internship project at Microsoft.

query [43, 44]. The backbone, therefore, does not have an explicit ability to localize, *i.e*., associate the given query to only the most relevant part of the long untrimmed video. As a resule, the cosine similarity between video features aligning and not aligning with the text query are indistinguishable, as shown in Fig 1b.

Inspired by the advantage shown in other tasks where pretraining and downstream setup match [3, 7, 23, 26], we hypothesize that formulating the pretraining itself as a VTG task on untrimmed videos can improve downstream grounding performance. The *untrimmed pretraining* will equip the model with a more accurate and fine-grained understanding of temporal boundaries within a given untrimmed video (as evidenced by the more precise predictions in blue in Fig. 1). We introduce ProTéGé, Untrimmed **Pr**etraining for Vide**o** **Te**mporal **G**rounding by Video T**e**mporal Grounding. ProTéGé is the first approach to formulate pretraining as a VTG task to bridge the gap between video backbones pretrained on trimmed videos and downstream VTG tasks working with untrimmed videos.

A critical challenge impeding this untrimmed pretraining is the scarcity of large-scale well-annotated video grounding datasets. There are, however, datasets such as HowTo100M [30] and Youtube-8M [1], with over a million untrimmed videos and corresponding subtitled text generated via automated speech-to-text APIs. One can potentially employ them for untrimmed pretraining as a VTG task. However, as noted by prior methods [12, 29, 41], since the text is derived from subtitles, the video regions are only noisily-correlated with the subtitled text, rendering the utility of these video-text pairs for grounding a non-trivial task.

To overcome the aforementioned challenges in leveraging large-scale untrimmed video datasets, we first propose a novel approach to transform them into VTG datasets. Then we introduce a novel video-text similarity grounding module along with an optimization objective that allows the pretraining to be robust to the noisy video-text correlations present in these datasets.

In this work, we use ProTéGé with HowTo100M in particular. To transform HowTo100M into a VTG dataset, ProTéGé introduces *aggregated subtitles* to concatenate one or more subtitles to form the text query and randomly samples an untrimmed video segment around the query. *Aggregated subtitles* allow ProTéGé to incorporate arbitrarily long text queries larger than the average 4s duration of a single subtitle. This way, we can synthesize millions of video-text grounding pairs for VTG pretraining. Using these pairs, ProTéGé performs pretraining with our novel Video-Text Similarity-based Grounding Module (VT-SGM). VT-SGM creates a 2D-proposal grid by computing the cosine similarity between the text query and the different temporal regions of the untrimmed video. It then learns to maximize the similarity between the query and the part that is most relevant to it. This is achieved via our novel pretraining objective that incorporates a distance-based localization loss which uses the noisy ground truth and a combination of inter-video and intra-video alignment losses. This allows the objective to balance the training via the noisy ground truth and multimodal video-text representation learning. We show that ProTéGé is very effective for VTG as a downstream task. It significantly outperforms backbones pretrained on trimmed videos on standard datasets across all variations of supervision. We summarize our contributions as,

1. We propose ProTéGé, the first pretraining method formulated as a video temporal grounding task to bridge the gap between pretraining and downstream video temporal grounding tasks in untrimmed videos.
2. We propose a novel algorithm including *aggregated subtitles*, a Video-Text Similarity-based Grounding Module, and a pretraining objective to leverage large-scale untrimmed video dataset HowTo100M with noisy video-text pairs.
3. Extensive experiments on standard datasets across multiple downstream tasks with different levels of supervision validate that our approach significantly improves the performance across all benchmarks.

2. Related Work

Video Temporal Grounding. First proposed in [2, 8], VTG aims to retrieve the temporal moment for the given sentence [46, 50]. Most works, including [2, 8, 20, 21, 32, 48, 51], are developed in a supervised setting, which utilizes the start and end timestamp as supervision. Some methods use reinforcement learning to address the problem [11, 13, 39]. Those methods formulate VTG as a sequential decision-making problem. Moreover, weakly supervised VTG is also proposed which only utilizes the sentence and video without any localization annotations, achieving competitive results [9, 19, 28, 31, 37]. Other works like PSVL [33] and DSCNet [22] proposed VTG in a zero-shot or unsupervised manner without seeing supervised information.

Video-Language Pre-training. Benefiting from multi-modality and large-scale datasets, significant works have studied pre-training tasks using both video and language [10, 14, 18, 27, 29, 34, 35, 40, 41, 47, 55]. Videobert [36] extracted a set of cooking videos from YouTube, and adapts the BERT model, learning a joint visual-linguistic representation. Clipbert [17] utilized image-text datasets for pre-training, and then fine-tune on the video-text downstream task. With the introduction of HowTo100M [30], a video dataset with more than 130M video clips and corresponding transcription, many works leverage the clip-caption pairs to learn joint text-video embedding. MIL-NCE [29] proposed a multiple-instance learning objective to deal with

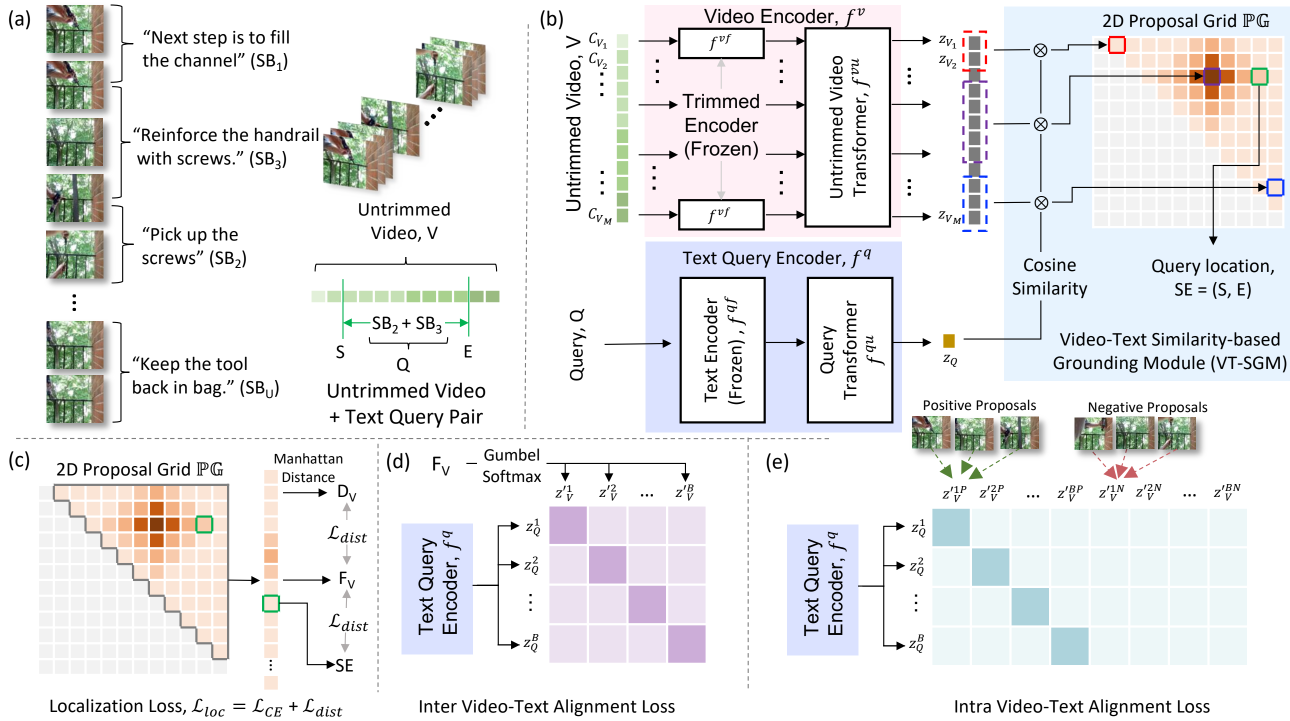

Figure 2. ProTéGé Overview: (a) We generate samples for VTG pretraining having untrimmed video V, text query Q, and query location $SE = (S, E)$. (b) We then process V and Q via the Video and Query encoder, respectively. We then transform video features, $\mathbf{z}_V$, into proposals and compute the cosine similarity of proposals with query features, z_Q, to get proposal grid $\mathbb{PG}$ via VT-SGM. Using SE, F_V (flattened $\mathbb{PG}$), $\mathbf{z}_V$, and z_Q, we perform pretraining via (c) Localization Loss and (d) Inter- and (e) Intra-Video-Text Alignment Loss.

the misaligned narration descriptions from Howto100M. Multimodal Pre-training(MMP) [14] extended Howto100M to a multilingual dataset to mitigate the performance gap degrade for non-English data. VLM [40] used a single BERT encoder to realize a task-agnostic pretraining, which can accept single/ multiple modalities for different downstream tasks. Videoclip [41] utilized temporally overlapped pairs to learn fine-grained associations between video frames and word tokens. TAN [12] introduced a temporal-alignment network that targeted refining alignable text in Howto100M [12]. Most existing pretraining works are focused on solving downstream tasks like text-video retrieval [2], VideoQA [42], video captioning [54] , and action recognition [4] under zero-shot or finetune settings.

Pre-training for localization. Several works study pretraining for localization tasks. Lofi [44] proposed to jointly optimize the video encoder from trimmed pretraining and TAL head, resolving the discrepancy problem for downstream tasks. PAL [49] designed a self-supervised pretext task for temporal action localization, achieving unsupervised pretraining. BSP [43] proposed a video synthesis method that used 4 different boundary strategies to generate videos with temporal boundary information to facilitate pretraining. LocVTP [5] combined coarse-grained and fine-grained contrastive loss and utilized a temporally aware contrastive learning to optimize the pretraining for temporal localization tasks. Although specifically designed for localization-related downstream tasks, those methods are still pretrained in a trimmed manner, which is sub-optimal for localization downstream tasks.

3. Method

Let $V = [v]_{i=1}^{T_V}$ be an arbitrarily long untrimmed video where $v_1, \dots, v_T$ denote the sequence of T_V frames forming the video. Let $Q = [q_i]_{i=1}^{L_Q}$ denote a text query comprising a sequence of L word tokens, $q_1, \dots, q_{L_Q}$. Since ProTéGé formulates the pretraining as an untrimmed VTG task to reduce the discrepancy between pretraining and downstream VTG, we formulate pretraining as localizing the start-end timestamp tuple (S, E) in the video V that best matches the textual description in query Q.

3.1. Synthesizing VTG Dataset for Pretraining

Since large-scale untrimmed video datasets with temporally well-annotated captions are unavailable for pretraining as a VTG task, we leverage the video dataset HowTo100M for VTG-based untrimmed pretraining in ProTéGé. HowTo100M comprises over a million untrimmed videos and autogenerated speech-to-text subtitles. The subtitles can potentially serve as text queries Q in our VTG-based untrimmed pretraining. On top of the challenge of noisy video-text correlation which we address in Sec 3.3, another key obstacle in leveraging such datasets is that the duration of a single subtitle text is very small (*e.g.*, 4s on avg. for HowTo100M) while downstream VTG datasets can have arbitrarily long captions (*e.g.*, 37s on avg. for ActivityNet-Captions [16]), leading to a discrepancy and sub-optimal performance.

We, therefore, propose to use *aggregated subtitles* where we concatenate one or more captions together to form the query. Specifically, for an untrimmed video $\mathcal{V}$, let $\mathbf{SB} = [SB_j]_{j=1}^{U}$ be the sequence of U subtitles. For the K^{th} sample of our synthesized VTG dataset, we form query Q_K as $Q_K = [SB_j]_{j=m}^{n} = [q_{m_1}, \ldots, q_{m_{L_{SB_m}}}, \ldots, q_{n_1}, \ldots, q_{n_{L_{SB_n}}}]$, where $1 \leq m \leq n \leq U$ and L_{SB_m} and L_{SB_n} are the number of word tokens in subtitles SB_m and SB_n respectively. Let v_s and v_e be the video frames where Q_K starts and ends respectively in $\mathcal{V}$. We also randomly sample an arbitrarily long video segment, $V_K = [v_j]_{j=s'}^{e'}$ with $v_{s'}$ and $v_{e'}$ as start and end frames of V_K respectively such that $1 \leq v_{s'} \leq v_s \leq v_e \leq v_{e'} \leq T_{\mathcal{V}}$. We then define the normalized start-end timestamp tuple (S_K, E_K) where Q_K temporally grounds in V_K as,

$$S_K, E_K = \frac{s - s'}{e' - s'}, \frac{e - s'}{e' - s'} \quad, 0 \leq S_K < Q_K \leq 1 \quad (1)$$

This enables us to synthesize dataset samples $\{V_K, Q_K, (S_K, E_K)\}$ for our VTG-based pretraining task (Fig 2a). It is worth noting that contrary to all previous works which leverage datasets like HowTo100M as an independent collection of trimmed video-text pairs where each pair is extracted using the subtitles' start-end timestamp, our proposed method to synthesize a VTG dataset makes it possible to leverage videos for pretraining as untrimmed with arbitrarily long duration.

3.2. Model Overview

To perform VTG during pretraining, ProTéGé consists of a network with a video encoder to extract features for the untrimmed video V_K and a text query encoder to encode the corresponding text query Q_K. We then use a novel video-text similarity-based grounding module that learns to align the text query with the most relevant part of the video via a novel pretraining objective for VTG. This enables learning features that are better suited for downstream VTG tasks. Fig 2 illustrates an overview of ProTéGé. We omit K in subsequent text for simplicity.

Video Encoder. As shown in Fig. 2b, ProTéGé comprises a video encoder $f^v(.)$ that takes as input an untrimmed video V with an arbitrary number of frames T_V. We design $f^v(.)$ to leverage the feature representations learned by the pretrained video encoder $f^{vf}(.)$ trained on a large set of trimmed videos. These features serve as a rich representation of the local temporal neighborhood and help reduce computational overhead by reusing already performed pretraining. Moreover, freezing $f^{vf}(.)$ allows $f^v(.)$ to focus on learning long-term temporal interactions in an untrimmed video, which is missing in $f^{vf}(.)$ and is the source of discrepancy between trimmed pretraining and downstream untrimmed VTG. We split arbitrarily long V into up to M clips, $\mathbf{C_V} = [C_{V_j}]_{j=1}^{M}$ where each clip has a fixed T_C number of frames (padding last clip with its last frame if needed). We first feed each clip to $f^{vf}(.)$ independently and do an average pooling over the frame dimension to obtain a sequence of clip-level local feature representations $\mathbf{h}_V = [h_{V_j}]_{j=1}^{M}$ of size $M \times D$ where D is the feature dimension. $\mathbf{h}_V$ is then fed to a transformer-based untrimmed video encoder, $f^{vu}(.)$, to obtain the temporal sequence of feature representations for the untrimmed video as $\mathbf{z}_V = [z_{V_j}]_{j=1}^{M}$ such that $\mathbf{z}_V = f^v(V) = f^{vu}(f^{vf}(V))$.

Query Encoder. To encode the text query Q, ProTéGé comprises a query encoder $f^q(.)$ whose design is symmetrical to the video encoder $f^v(.)$ (Fig 2b). $f^q(.)$ comprises a frozen pre-trained text encoder $f^{qf}(.)$ which first tokenizes the query text via embedding lookup and then processes it to output a sequence of features $\mathbf{h_Q}$. After this, we feed $\mathbf{h_Q}$ through a randomly initialized transformer-based query encoder $f^{qu}(.)$ which performs an average pooling over the features for each at the end to obtain the final query feature embedding $z_Q = f^q(Q) = \text{AvgPool}(f^{qu}(f^{qf}(Q)))$.

Video-Text Similarity-based Grounding Module (VT-SGM). Similar to downstream VTG tasks, we formulate VTG during pretraining as a task to align text with the correct region in the untrimmed video. This explicitly primes the model to specialize for downstream VTG tasks. Unlike video pretraining methods that leverage trimmed videos, we cannot align the text features with the entire video feature sequence. To tackle this, we propose a novel Video-Text Similarity-based Grounding Module (VT-SGM) that learns to localize the text query in the untrimmed video. The module first generates an upper triangular 2D proposal grid, $\mathbb{PG} \in \mathbb{R}^{M \times M}$, where each grid cell maps to a video proposal (Fig 2b). Taking output $\mathbf{z}_V = [z_{V_j}]_{j=1}^{M}$ of video encoder $f^v(.)$, a cell, pg_{se} at row s and column e in proposal grid $\mathbb{PG}$ maps to a video proposal spanning the feature sequence $[z_{V_s}, \ldots, z_{V_e}]$ where $1 \leq s \leq e \leq M$. We obtain the feature for video proposal for each cell, $\mathbf{z}_{vp_{se}}$ by applying an average pooling over the temporal dimension such that $\mathbf{z}_{vp_{se}} = \frac{\sum_{t=s}^{e} z_{V_t}}{e-s-1}$ $\forall 1 \leq s \leq e \leq M$. Finally, we compute the grid score g_{se} of each cell pg_{se}, as cosine similarity of the text query features z_Q, obtained from the query encoder $f^q(.)$, with $\mathbf{z}_{vp_{se}}$ for every $1 \leq s \leq e \leq M$ as,

$$g_{(s,e)} = \frac{\mathbf{z}_{vp_{se}}^T z_Q}{\|\mathbf{z}_{vp_{se}}\| \|z_Q\|} \quad (2)$$

The similarity score g_{se} reflects how well the region in the untrimmed video represented by $\mathbf{z}_{vp_{se}}$ aligns with the query text feature z_Q. We take inspiration for the design of $\mathbb{PG}$ from 2D-TAN [51] but we differ significantly in that unlike 2D-TAN, $\mathbb{PG}$ is a 2D grid instead of a 3D temporal feature grid used by 2D-TAN. We use cosine similarity to obtain the grid scores whereas 2D-TAN applies elementwise multiplication on video-text features followed by a series of convolutional layers to obtain the final grid scores. We find our approach less complex and therefore more robust (Sec 4) in the presence of data from sources like HowTo100M with noisy video-text correlations.

3.3. Pretraining Objective for Temporal Grounding

One major challenge in leveraging the synthesized VTG dataset from HowTo100M (Sec 3.1) is that the subtitles may not necessarily align with the video.To enable our model to be robust to these noisy video-text correlations, we design a novel pretraining objective for VTG which augments what the model learns from the noisy ground truth $SE = (S, E)$ with what the model can learn implicitly via multimodal video-text representation learning. Our pretraining objective, therefore, comprises a *localization loss*, *inter-video-text alignment loss*, and *intra-video-text alignment loss*.

Localization Loss. The localization loss, $\mathcal{L}_{loc}$ leverages the fact that although our ground truth $SE = (S, E)$ is noisy, it can still provide some guidance to the model in terms of approximately what part of the untrimmed video can be best described by the text query. Therefore, one component of this loss is the standard cross-entropy loss, $\mathcal{L}_{ce} = \text{CrossEntropyLoss}(F_V, SE)$, where $F_V \in \mathbb{R}^H$ is the sequence of proposal grid scores $g_{se}\ \ \forall 1 \leq s \leq e \leq M$ obtained by flattening the 2D proposal grid, $\mathbb{PG}$ and H is the total number of video proposals, $H = \frac{M(M+1)}{2}$ (Fig 2c).

$\mathcal{L}_{ce}$ serves as a hard-localization loss which does not account for the proximity of the localization from the ground truth and equally penalizes all incorrect localizations. $\mathcal{L}_{ce}$ alone is not optimal in the presence of noisy ground truth that we have. We, therefore, propose an additional component of $\mathcal{L}_{loc}$ that penalizes an incorrect prediction proportionate to how far is the localization prediction from the ground truth. To achieve this soft localization, we compute a distance map, $\mathbb{D} \in \mathbb{R}^{M \times M}$, such that $D(s, e)$ is defined as the Manhattan distance between the proposal corresponding to (s, e) and the ground truth $SE = (S, E)$, $D(s, e) = |S - s| + |E - e|$. Similar to $\mathcal{L}_{ce}$, we flatten $\mathbb{D}$ to a sequence of distance scores $D_V \in \mathbb{R}^H$ and compute the distance-based soft-localization loss, $\mathcal{L}_{dist}$ as the L1 loss,

$$\mathcal{L}_{dist} = \left\| F_V, 2\left(1 - \frac{D_V - \min(D_V)}{\max(D_V) - \min(D_V)}\right) - 1 \right\|_1 \quad (3)$$

The second term transforms the distance scores to be scaled to $(-1, 1)$ so that their range is the same as the cosine similarity scores in F_V. Therefore, when the second term is close to 1, it represents that the localization prediction is closer to the ground truth while the second term being close to -1 represents that the prediction is farthest possible from the ground truth. $\mathcal{L}_{dist}$, therefore, applies a softer distance-relative localization constraint and encourages the proposals closer to the ground truth to obtain higher similarity scores compared to those farther away from the ground truth. We define the final localization loss as $\mathcal{L}_{loc} = \mathcal{L}_{ce} + \mathcal{L}_{dist}$

Inter-Video-Text Alignment Loss We argue that the localization loss, $\mathcal{L}_{loc}$, alone is not adequate to mitigate the negative impact of noisy correlations as the loss may still put a considerable emphasis on the ground truth SE to guide the optimization. We, therefore, take inspiration from trimmed video-text pretraining methods [29, 41, 47] to propose an Inter-Video-Text Alignment Loss, $\mathcal{L}_{inter}$. $\mathcal{L}_{inter}$ maximizes the alignment of text query Q with the video V across the alignments of Q with all other videos in the training batch, B (Fig 2d). However, unlike trimmed video-text pretraining, defining a video-level representation to align with query $\mathcal{L}_{inter}$ is non-trivial because we cannot align the query with a globally averaged video representation. We, therefore, propose to leverage the similarity scores from the flattened proposal grid similarity scores F_V to compute a weighted proposal that best aligns with the query. Then, we do an average pooling on this weighted proposal to define a video-level representation for the untrimmed video as,

$$z'_V = \text{AvePool}(Z_V \cdot \text{Gumbel\_Softmax}(F_V)) \quad (4)$$

where Z_V are the video proposal features corresponding to F_V obtained by flattening $\mathbf{z}_{vp_{se}}$ in $\mathbb{PG}$. The Gumbel_Softmax sets a larger weight to proposals which have higher cosine similarity with the query. Using z'_V and z'^{+}_V, z'^{-}_V as positive/negative video pairings,

$$\mathcal{L}_{inter} = -\sum_{(z'_V, z_Q) \in B} \left(\log \frac{\exp(z_Q \cdot z'^{+}_V / \tau)}{\sum_{z \in \{z'^{+}_V, z'^{-}_V\}} \exp(z_Q \cdot z / \tau)} \right) \quad (5)$$

The above formulation affords two benefits, (1) since we use similarity scores to obtain z'_V without involving ground truth, it enables $\mathcal{L}_{inter}$ to guide the network to focus on a better video region candidate based on what the model has learned over training in case the ground truth for a certain sample K is noisy. (2) the Gumbel_Softmax function keeps the weighted proposal differentiable to allow influencing the features from the entire untrimmed video.

Intra-Video-Text Alignment Loss While $\mathcal{L}_{inter}$ enforces a better alignment of the video with the query, it may still cause the model to not focus on the most representative region of the video. For this, we need a second alignment loss to increase the margin between the video regions that align better with the query and those that align poorly. Inspired by weakly-supervised grounding methods [53], we define Intra-Video-Text Alignment Loss $\mathcal{L}_{intra}$ as,

$$\mathcal{L}_{intra} = -\sum_{(z'_V, z_Q)\in B}\left(\log \frac{\exp(z_Q \cdot z'^P_V/\tau)}{\sum_{z\in\{z'^P_V, z'^N_V z'^-_V\}}\exp(z_Q\cdot z/\tau)}\right) \quad (6)$$

where z'^P_V and z'^N_V are video features obtained by averaging P and N are positive and negative proposal sets containing the top-pr proposals with the highest and the lowest video-text similarity scores respectively (Fig 2e).

We optimize ProTéGé using a combination of the above three losses defined as $\mathcal{L}_{VTG} = \mathcal{L}_{loc} + w_1 \cdot \mathcal{L}_{inter} + w_2 \cdot \mathcal{L}_{intra}$ where w_1 and w_2 are loss coefficients.

4. Experiments

Datasets. We evaluate ProTéGé on common VTG datasets, Charades-STA [8] and ActivityNet-Captions [16], using their standard splits. The former is built on the Charades dataset containing 9,848 videos of daily indoor activity scenarios with 12,408 and 3,720 video-sentence pairs in the training and test set, respectively. The latter is built on ActivityNet v1.3, which contains 20k YouTube videos, with 37,417, 17,505, and 17,031 video-sentence pairs in training, val_1, and val_2 set respectively.

Implementation Details. We synthesize our pretraining VTG dataset using HowTo100M [30] with 1.22M untrimmed videos. To obtain text queries Q, we randomly sample *aggregated subtitles* from a video up to a max duration of 50 secs. The duration of untrimmed video segment V_K is up to 128s, randomly sampled around Q at 30fps. V_K is split into clips of $T_C = 64$ frames each to obtain an arbitrary number of clips up to $M = 60$ (applying an attention mask for videos shorter than 128s). We use frozen pretrained Video Swin Transformer [25] for trimmed video encoder $f^{vf}(.)$, pretrained Roberta [24] for text encoder $f^{qf}(.)$, and 4-layer BERT [6] for $f^{vu}(.)$ and $f^{qu}(.)$. We train on 32 NVIDIA P100 GPUs with 2048 batch size for 40 epochs, SGD optimizer, effective LR of 0.2 decayed via cosine scheduler, $w_1 = 1$, $w_2 = 1$, $\tau = 1$, and $pr = 3$. Please refer to the supplementary for more details.

Downstream Tasks. To validate the effectiveness of ProTéGé, we show results on video temporal grounding task with three representative settings: fully supervised temporal grounding, weakly supervised temporal grounding and zero-shot temporal grounding.

4.1. Fully-supervised Video Temporal Grounding

We show how the pretrained features from ProTéGé improve the downstream task of supervised video temporal grounding. Since the task is fully-supervised, it assumes that both the query and start/end timestamps are available during training. We demonstrate this task using 2D-TAN as the downstream method for consistent comparison across different pretrained video backbones using their respective pretrained visual features. Following 2D-TAN, we use Top-1 Recall at 0.5 and 0.7 tIoU thresholds for comparison.

Table 1a and b summarize the results for Charades-STA and ActivityNet-Captions respectively. From Table 1a, we can observe that compared to Swin-B and Swin-T video backbones which are pretrained on trimmed videos, ProTéGé, which is pretrained on untrimmed videos, can achieve significantly high improvement of 7.15%/5.6% and 4.20%/4.71% respectively on R@0.5/R@0.7 metrics on Charades-STA. We also outperform similarly on ActivityNet-Captions as shown in Table 1b. Moreover, we compare ProTéGé using Swin-T backbone with existing methods doing pretraining on trimmed video-text pairs with backbones of comparable size (Row 1-5,10 in Table 1a and Row 1-5,9 in Table 1b). ProTéGé significantly outperforms all baselines by at least 6.29%/5.13% and 1.02%/1.75% on R@0.5/R@0.7 metrics on Charades-STA and ActivityNet-Captions respectively. Further, ProTéGé using Swin-B outperforms LocVTP of comparable size by 9.66%/4.08% on R@0.5/R@0.7 metrics. We cannot compare ProTéGé with LocVTP on ActivityNet-Caption as LocVTP seems to have val_1 split in the training set which makes test videos part of training and artificially elevates the scores.

These results validate the effectiveness of ProTéGé and the significance of pretraining on untrimmed videos via VTG in achieving higher performance on downstream VTG in a supervised setting. Moreover, we achieve a higher performance of 4.17%/2.00% using Swin-B than Swin-T which also highlights the capability of ProTéGé to scale with more data and network parameters.

4.2. Weakly-supervised Video Temporal Grounding

We assess the effectiveness of the pretrained features from ProTéGé on weakly-supervised VTG task. This task assumes that while the query is available during training, we do not have access to its location in the video in the form of a start-end timestamp. We choose CPL on Charades-STA and CNM on ActivityNet-Captions which achieve state-of-the-art (SoTA) performance on the respective datasets. Following CPL and CNM, we report Top-1 Recall at tIoU threshold 0.3, 0.5, 0.7 for Charades-STA and 0.1, 0.3, 0.5 for ActivityNet-Captions as well as mIoU metric.

Table 1c tabulates the results for Charades-STA. We report results of CPL on Swin-T and Swin-B for a fair comparison. From the table, we can observe that using Swin-T and Swin-B with CPL, our method pretrained on untrimmed videos outperforms corresponding video backbones pretrained on trimmed videos by 1.58% and 0.98% respectively on mIoU metric. ProTéGé exceeds all other baselines by at least 3.17% and 4.82% using Swin-T and Swin-B respectively, again validating that it can achieve even higher performance with larger-scale backbones. Table 1e shows the comparison on ActivityNet-Captions. Comparing ProTéGé

Table 1. Fully-supervised Video Temporal Grounding using 2D-TAN on (a) Charades-STA and (b) ActivityNet-Captions. Weakly-supervised Video Temporal Grounding on (c) Charades-STA and (e) ActivityNet-Captions. (d) Zero-shot Video Temporal Grounding on Charades-STA and Activity-Caption. Using pretrained features from ProTéGé shows significant improvements over features from all existing trimmed pretrained backbones over all benchmarks. †reported in paper, ‡obtained via official model weights

Model	R@0.5	R@0.7
VideoBert [36]	32.7	19.5
MIL-NCE [29]	37.0	21.2
UniVL [27]	38.2	22.7
SupportSet [34]	37.4	21.6
LocVTP [5]	43.6	26.3
2D-TAN [51]	42.80	23.25
2D-TAN(w/ Swin-T)	44.89	23.87
2D-TAN(w/ Swin-B)	46.11	24.68
Ours (w/ Swin-T)	**49.09**	**28.38**
Ours (w/ Swin-B)	**53.26**	**30.38**

(a)

Model	R@0.5	R@0.7
VideoBert [36]	37.2	21.0
MIL-NCE [29]	41.8	24.5
UniVL [27]	42.2	25.4
SupportSet [34]	41.9	25.2
2D-TAN [51]	44.51	26.54
2D-TAN(w/ Swin-T)	43.42	25.74
2D-TAN(w/ Swin-B)	44.78	26.51
Ours (w/ Swin-T)	**45.53**	**28.29**
Ours (w/ Swin-B)	**45.99**	**29.02**

(b)

Method	R@0.3	R@0.5	R@0.7	mIoU
LCNet [45]	59.60	39.19	18.87	38.94
WSTAN [38]	43.39	29.35	12.28	-
RTBPN [52]	60.04	32.36	13.24	-
CNM [53]	60.04	35.15	14.95	38.11
CPL (w/ Swin-T) [53]	61.68	44.61	20.61	40.53
CPL (w/ Swin-B) [53]	65.99	47.97	21.53	42.78
CPL + Ours (w/ Swin-T)	63.93	46.89	21.37	**42.11**
CPL + Ours (w/ Swin-B)	67.44	49.33	22.16	**43.76**

(c)

Dataset	Method	R@0.3	R@0.5	R@0.7	mIoU
Charades-STA	PSVL† [33]	46.47	31.29	14.17	31.24
	PSVL‡ [33]	46.63	30.84	13.57	30.09
	PSVL + Ours	46.79	31.84	17.51	**31.25**
ActivityNet-Captions	PSVL† [33]	44.74	30.08	14.74	29.62
	PSVL‡ [33]	43.03	25.17	10.98	30.78
	PSVL + Ours	45.02	27.85	14.89	**33.04**

(d)

Method	R@0.1	R@0.3	R@0.5	mIoU
LCNet [45]	78.58	48.49	26.33	34.29
WSTAN [38]	79.78	52.45	30.01	-
RTBPN [52]	73.73	49.77	29.63	-
CPL [53]	82.55	55.73	31.37	36.82
CNM (w/ CLIP) [53]	78.13	55.68	33.33	37.14
CNM + Ours (w/ Swin-T)	78.42	56.16	35.27	**38.10**
CNM + Ours (w/ Swin-B)	81.70	59.49	38.18	**40.29**

(e)

using Swin-T with CNM using CLIP, it achieves a 0.96% higher mIoU. This is noteworthy as Swin-T is a smaller backbone than CLIP. When using Swin-B, ProTéGé is able to achieve 3.15% higher mIoU. It also performs at least 3.47% higher than all other previous methods. These results further validate the effectiveness of our untrimmed pretraining via VTG on downstream VTG tasks. This also demonstrates the versatility of ProTéGé in improving VTG performance on scenarios beyond fully-supervised setting.

4.3. Zero-shot Video Temporal Grounding

To further demonstrate ProTéGé's ability to be effective on VTG tasks with varying degrees of supervision, we show the performance of using pretrained features from ProTéGé on zero-shot video temporal grounding. This downstream task assumes that during training, neither the text query nor the corresponding start-end timestamp is available. We demonstrate this task using PSVL on Charades-STA and ActivityNet-Captions and report results using Top-1 Recall at tIoU thresholds of 0.3, 0.5, and 0.7.

Table 1d shows the results on Charades-STA and ActivityNet-Captions respectively. For a fair comparison, we compare with results from PSVL obtained via the officially provided model weights but we also provide the results reported by PSVL in the paper for reference. We use Swin-T for ProTéGé's experiments as it is a comparable backbone. We can observe that ProTéGé significantly outperforms PSVL with a mIoU improvement of 1.14% on Charades-STA and 2.26% on ActivityNet-Captions. Improvements from ProTéGé on this benchmark further establish the usefulness of performing untrimmed pretraining via VTG to benefit all types of downstream VTG tasks.

4.4. Ablation Study

To evaluate the contribution of each novel component of ProTéGé, we conduct an ablation study. We experiment on Charades-STA using Swin-B backbone on fully-supervised VTG and report results in Table 2a. We first observe that removing the localization loss, $\mathcal{L}_{loc}$, from our pretraining objective leads to a drop in performance by 3.86%/3.78% on the R@0.5/R@0.7 metric. This shows that even though the video-text correlations are noisy, it is still important to leverage the ground truth SE via a combination of $\mathcal{L}_{CE}$ and $\mathcal{L}_{dist}$ to perform effective pretraining. Next, we observe that if we instead remove the video-text alignment loss, $\mathcal{L}_{inter}$ and $\mathcal{L}_{intra}$, from the pretraining objective, it reduces the performance by 1.63%/0.79%. This helps to validate that without the alignment losses and fully relying on the localization loss is not optimal as it makes the module more vulnerable to the noisy video-text correlations due to increased dependence on the ground truth SE.

Table 2. (a) Ablation study showing each component of ProTéGé plays a significant role in achieving the optimal performance. (b) Analysis on the number of top-pr proposals used in $\mathcal{L}_{intra}$

Method	R@0.5	R@0.7
Ours	**53.26**	**30.38**
- w/o $\mathcal{L}_{loc}$	49.40	26.60
- w/o $\mathcal{L}_{inter} + \mathcal{L}_{intra}$	51.63	29.59
- w/o VT-SGM	48.36	25.22
- w/o Untrimmed	47.35	25.47

$\mathcal{L}_{intra}, pr$	R@0.5	R@0.7
2	52.26	29.48
3	**53.26**	**30.38**
4	53.26	30.35
5	52.50	29.34

We also explore the contribution of our novel grounding module, VT-SGM, towards better pretraining. We conduct an experiment where we replace our 2D proposal grid with a regression layer that directly predicts the start and end timestamp for grounding. We find this setting to perform significantly worse with a drop of 4.9%/5.16%. This proves that given the imperfect video-text correlations in the pretraining dataset, direct regression is prone to noise and our VT-SGM allows for a softer localization to learn better grounding-oriented feature representations. We further validate the importance of pretraining on untrimmed videos where we directly perform video-text alignment using our model backbone. This performs 5.91%/4.91% worse than ProTéGé, highlighting the merit of pretraining on untrimmed videos.

Table 3. Analysis of different loss functions showing that each loss contributes significantly towards the optimal performance

Method	R@0.5	R@0.7
Ours	**53.26**	**30.38**
Localization loss		
- w/o $\mathcal{L}_{CE}$	51.18	27.33
- w/o $\mathcal{L}_{dist}$	51.86	28.35
- w/o $\mathcal{L}_{CE} + \mathcal{L}_{dist}$	49.40	26.60
Alignment loss		
- w/o $\mathcal{L}_{intra}$	52.45	29.22
- w/o $\mathcal{L}_{inter}$	52.39	28.88
- w/o $\mathcal{L}_{intra} + \mathcal{L}_{inter}$	51.63	29.59

4.5. Discussion

We assess the different components of ProTéGé in detail to understand their influence on pretraining and downstream performance. We experiment on Charades-STA using the Swin-B backbone on fully-supervised VTG.

Pretraining Objective. Table 3 reports the performance for different combinations of losses in our pretraining objective. For localization loss without cross-entropy, $\mathcal{L}_{CE}$, the performance drops by 2.08%/3.05% while removing the distance loss, $\mathcal{L}_{dist}$, leads to a reduction of 1.4%/2.03%. This highlights that both $\mathcal{L}_{CE}$ and $\mathcal{L}_{dist}$ contribute significantly in the pretraining objective. Either soft-localization via $\mathcal{L}_{dist}$ or hard-localization via $\mathcal{L}_{CE}$ alone causes the model to under-utilize the available ground truth SE information in the synthesized VTG pretraining dataset. We similarly find removing the inter-alignment loss, $\mathcal{L}_{inter}$ to reduce performance by 0.87%/1.5% and removing intra-alignment loss, $\mathcal{L}_{intra}$ to reduce performance by 0.81%/1.16%. This validates that both video-text alignment losses contribute to the best performance of ProTéGé. Both losses leverage representation learning to make the model robust to noisy ground truth; $\mathcal{L}_{inter}$ leverages it via alignment and $\mathcal{L}_{intra}$ uses it to make features within a video more discriminative.

Top-pr proposals in $\mathcal{L}_{intra}$. Table 2b further shows the results on selecting a different number of proposals as part of $\mathcal{L}_{intra}$. We find that using 3 proposals gives the optimal performance with 4 proposals being comparable. Having 2 or 5 proposals leads to worse performance. We believe having fewer proposals can cause the model to miss relevant proposals while having a large number of proposals averages out the relevance of the high-ranking proposals.

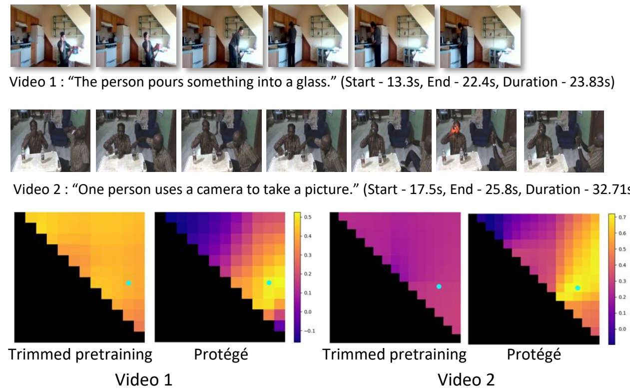

Figure 3. Visualization of 2D proposal grid on unseen Charades-STA video. ProTéGé features show higher variation in cosine similarity with larger similarity closer to the ground truth (cyan dot).

Zero-shot visualization of 2D grid. Fig 3 compares the similarity score proposal grid on a video from the Charades-STA dataset when our model is pretrained on untrimmed videos and trimmed videos. The model has never seen this video. We feed the video and query through our model and obtain the cosine similarity scores from the 2D proposal grid without doing any finetuning on the video. The cyan dot in the grid denotes the ground truth proposal for the corresponding query. We can observe that when training on untrimmed video, our method can clearly learn to exhibit higher similarity close to the ground truth and lower similarity farther away from the ground truth. But when trained on trimmed videos, there is no visible difference in the similarity scores across proposals, highlighting the importance of pretraining on untrimmed videos to learn fine-grained discriminative features within a video.

5. Conclusion

We present ProTéGé as the first method to bridge the gap between pretraining and downstream VTG by pretraining on untrimmed videos via VTG. To do so, ProTéGé first synthesizes a VTG pretraining dataset from large-scale video dataset HowTo100M with noisy video-text pairs using *aggregated subtitles* and then performs pretraining via a novel Video-Text Similarity-based Grounding Module (VT-SGM) and pretraining objective comprising a localization loss and inter- and intra-video-text alignment losses. Extensive experiments validate that pretrained features from ProTéGé significantly improve the performance on downstream VTG with full, weakly, and zero-shot training supervision.

Acknowledgements. At MSU, Lan Wang is supported in part by award 60NANB18D210 by NIST, U.S. Dept. of Commerce.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 2

[2] 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/CVF International Conference on Computer vision*, 2017. 2, 3

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

[4] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. 3

[5] Meng Cao, Tianyu Yang, Junwu Weng, Can Zhang, Jue Wang, and Yuexian Zou. Locvtp: Video-text pre-training for temporal localization. In *European Conference on Computer Vision*, pages 38–56. Springer, 2022. 3, 7

[6] 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. 6

[7] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*, 2020. 2

[8] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE/CVF International Conference on Computer vision*, 2017. 2, 6

[9] Mingfei Gao, Larry S Davis, Richard Socher, and Caiming Xiong. Wslln: Weakly supervised natural language localization networks. *arXiv preprint arXiv:1909.00239*, 2019. 2

[10] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. Coot: Cooperative hierarchical transformer for video-text representation learning. *Advances in neural information processing systems*, 2020. 2

[11] Meera Hahn, Asim Kadav, James M Rehg, and Hans Peter Graf. Tripping through time: Efficient localization of activities in videos. *arXiv preprint arXiv:1904.09936*, 2019. 2

[12] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2906–2916, 2022. 2, 3

[13] Dongliang He, Xiang Zhao, Jizhou Huang, Fu Li, Xiao Liu, and Shilei Wen. Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. 2

[14] Po-Yao Huang, Mandela Patrick, Junjie Hu, Graham Neubig, Florian Metze, and Alexander Hauptmann. Multilingual multimodal pre-training for zero-shot cross-lingual transfer of vision-language models. *arXiv preprint arXiv:2103.08849*, 2021. 2, 3

[15] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[16] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017. 4, 6

[17] 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*, 2021. 2

[18] 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. 2

[19] Zhijie Lin, Zhou Zhao, Zhu Zhang, Qi Wang, and Huasheng Liu. Weakly-supervised video moment retrieval via semantic completion network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2

[20] Daizong Liu, Xiaoye Qu, Jianfeng Dong, Pan Zhou, Yu Cheng, Wei Wei, Zichuan Xu, and Yulai Xie. Context-aware biaffine localizing network for temporal sentence grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[21] Daizong Liu, Xiaoye Qu, Xiao-Yang Liu, Jianfeng Dong, Pan Zhou, and Zichuan Xu. Jointly cross-and self-modal graph attention network for query-based moment localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2020. 2

[22] Daizong Liu, Xiaoye Qu, Yinzhen Wang, Xing Di, Kai Zou, Yu Cheng, Zichuan Xu, and Pan Zhou. Unsupervised temporal video grounding with deep semantic clustering. *arXiv preprint arXiv:2201.05307*, 2022. 2

[23] Song Liu, Haoqi Fan, Shengsheng Qian, Yiru Chen, Wenkui Ding, and Zhongyuan Wang. Hit: Hierarchical transformer with momentum contrast for video-text retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11915–11925, 2021. 2

[24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 6

[25] 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. 6

[26] 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*, 2019. 2

[27] 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, 7
[28] Minuk Ma, Sunjae Yoon, Junyeong Kim, Youngjoon Lee, Sunghun Kang, and Chang D Yoo. Vlanet: Video-language alignment network for weakly-supervised video moment retrieval. In *European conference on computer vision*, 2020. 2
[29] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2, 5, 7
[30] 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*, 2019. 2, 6
[31] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2
[32] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2
[33] Jinwoo Nam, Daechul Ahn, Dongyeop Kang, Seong Jong Ha, and Jonghyun Choi. Zero-shot natural language video localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1470–1479, 2021. 2, 7
[34] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, Joao Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. *arXiv preprint arXiv:2010.02824*, 2020. 2, 7
[35] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019. 2
[36] 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*, 2019. 2, 7
[37] Reuben Tan, Huijuan Xu, Kate Saenko, and Bryan A Plummer. Logan: Latent graph co-attention network for weakly-supervised video moment retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2083–2092, 2021. 2
[38] Yuechen Wang, Jiajun Deng, Wengang Zhou, and Houqiang Li. Weakly supervised temporal adjacent network for language grounding. *IEEE Transactions on Multimedia*, 2021. 7
[39] Jie Wu, Guanbin Li, Si Liu, and Liang Lin. Tree-structured policy based progressive reinforcement learning for temporally language grounding in video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2
[40] Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Masoumeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. Vlm: Task-agnostic video-language model pre-training for video understanding. *arXiv preprint arXiv:2105.09996*, 2021. 2, 3
[41] 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. 2, 3, 5
[42] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 3
[43] Mengmeng Xu, Juan-Manuel Pérez-Rúa, Victor Escorcia, Brais Martinez, Xiatian Zhu, Li Zhang, Bernard Ghanem, and Tao Xiang. Boundary-sensitive pre-training for temporal localization in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2, 3
[44] Mengmeng Xu, Juan Manuel Perez Rua, Xiatian Zhu, Bernard Ghanem, and Brais Martinez. Low-fidelity video encoder optimization for temporal action localization. *Advances in Neural Information Processing Systems*, 2021. 2, 3
[45] Wenfei Yang, Tianzhu Zhang, Yongdong Zhang, and Feng Wu. Local correspondence network for weakly supervised temporal sentence grounding. *IEEE Transactions on Image Processing*, 30:3252–3262, 2021. 7
[46] Yitian Yuan, Xiaohan Lan, Xin Wang, Long Chen, Zhi Wang, and Wenwu Zhu. A closer look at temporal sentence grounding in videos: Dataset and metric. In *Proceedings of the 2nd International Workshop on Human-centric Multimedia Analysis*, 2021. 2
[47] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387, 2022. 2, 5
[48] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2
[49] Can Zhang, Tianyu Yang, Junwu Weng, Meng Cao, Jue Wang, and Yuexian Zou. Unsupervised pre-training for temporal action localization tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3
[50] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. The elements of temporal sentence grounding in videos: A survey and future directions. *arXiv preprint arXiv:2201.08071*, 2022. 2
[51] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment local-

ization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2, 5, 7

[52] Zhu Zhang, Zhijie Lin, Zhou Zhao, Jieming Zhu, and Xiuqiang He. Regularized two-branch proposal networks for weakly-supervised moment retrieval in videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2020. 7

[53] Minghang Zheng, Yanjie Huang, Qingchao Chen, Yuxin Peng, and Yang Liu. Weakly supervised temporal sentence grounding with gaussian-based contrastive proposal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 5, 7

[54] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 3

[55] 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*, 2020. 2

Supplementary Material
ProTéGé: Untrimmed Pretraining for Video Temporal Grounding by Video Temporal Grounding

Lan Wang[⋆‡] Gaurav Mittal[⋆†] Sandra Sajeev[†] Ye Yu[†] Matthew Hall[†]
Vishnu Naresh Boddeti[‡] Mei Chen[†]
[†]Microsoft [‡]Michigan State University
{gaurav.mittal, yu.ye, mathall, ssajeev, mei.chen}@microsoft.com
{wanglan3,vishnu}@msu.edu

Below we provide additional qualitative and quantitative analysis for ProTéGé that we could not include in the main paper due to space constraints but was ready at the time of submission. Sec 1 provides a further discussion on hyper-parameter selection related to the maximum duration of text query Q, the maximum number of video clips M, and batch size B used during untrimmed pretraining. Next, Sec 2 provides visual analysis for ProTéGé both for pretraining and downstream Video Temporal Grounding (VTG), and finally, Sec 3 provides additional implementation details.

1. Additional Discussion

Maximum duration of text query Q. Since our method is not limited to a single subtitle as text query and uses *aggregated subtitles* by concatenating them, we discuss the effect of maximum query length in terms of duration in Table 1. We can observe that a maximum query duration of 50 seconds gives the best performance. We believe shorter durations limit the generalization of the method to downstream tasks with diverse query sizes. Meanwhile, a duration of 100 seconds is sub-optimal because by then, the query has too much information by having as many as 25 subtitles. This reduces its usefulness to precisely localize and be associated with a particular video segment.

Table 1. Effect of maximum duration of text query Q.

Max duration of Q (s)	R@0.5	R@0.7
10	52.59	27.48
20	50.19	26.83
50	53.26	30.38
100	48.27	27.73

Maximum number of video clips M. The video duration, in terms of the number of clips M, can play a significant role in the performance. As shown in Table 2, using at most $M = 15$ or $M = 30$ video clips significantly reduces R@0.5 by 5.28% and 7.07% respectively compared to using $M = 60$. This suggests that using long untrimmed videos makes the pretrained features more favorable for downstream VTG tasks. We also find that increasing M from 60 to 120 does not provide a significant improvement. Larger M results in more fine-grained proposals which are quadratically more in number. We believe that this increases task complexity making the training more challenging while also requiring longer training time as well as higher GPU memory. So for compute efficiency, we use $M = 60$ in our experiments.

Table 2. Effect of maximum number of video clips M.

Max video clips M	R@0.5	R@0.7
15	46.19	25.14
30	47.98	25.59
60	53.26	30.38
120	53.24	30.60

Table 3. Effect of batch size B during pretraining.

Batch size B	R@0.5	R@0.7
512	50.67	29.11
1024	52.51	30.86
2048	53.26	30.38
4096	51.74	28.69

[⋆]Authors with equal contribution.
This work was done as Lan Wang's internship project at Microsoft.

Batch size B during pretraining. The batch size B during untrimmed pretraining of ProTéGé decides the number of negative samples in $\mathcal{L}_{inter}$ and influences model training. Table 3 shows the results for using different batch sizes for pretraining. We find that using $B = 2048$ gives the overall best performance and having smaller or larger batch size leads to worse performance. We believe that having a smaller batch size can cause the model to have an insufficient number of negative samples while having a larger batch size could lead to a high number of false negatives. Both of these scenarios can impede model training [5].

Evaluation on VT-SGM. The proposed grounding module (VT-SGM) is directly incorporated into pretraining to leverage untrimmed videos for VTG. While downstream VTG methods like 2D-TAN inspire VT-SGM, our module's design is tailored to perform VTG-based untrimmed pretraining. To illustrate the effectiveness of VT-SGM, we replace VT-SGM in ProTéGé with original 2D-TAN and finding that it leads to 2.7%/4.0% lower R@0.5/R@0.7 on Charades and 1.8%/1.8% lower R@0.3/R@0.5 on TACoS, as shown in Table 4, further showing the benefit of the novel design.

Table 4. Evaluation on VT-SGM.

	Charades-STA		TACoS	
Grounding Module	R@0.5	R@0.7	R@0.3	R@0.5
VT-SGM	**53.26**	**30.38**	**43.63**	**31.39**
2D-TAN	50.53	26.40	41.88	29.36

Evaluation on more datasets. We show video temporal grounding evaluation on more datasets. Table 6 and 5 shows results on TACoS and QVHighlights using ProTéGé with Moment-DETR and 2D-TAN as downstream methods. It can be observed that untrimmed pretraining (Row 3) leads to 2.1%/4.2% better R@0.5/R@0.7 on QVHighlights and 1.8%/1.2% better R@0.3/R0.5 on TACoS vs. trimmed pretraining (Row 2) using the same extra data, which empirically validates the our untrimmed pretraining model. Moreover, comparing with the baselines, ProTéGé also shows superior performance.

Table 5. Evaluation on QVhighlight.

Method	R@0.5	R@0.7
Moment-DETR [1]	53.94	34.84
ProTéGé w/o Untrimmed	53.53	31.91
ProTéGé	**55.56**	**36.11**

Table 6. Evaluation on TACoS.

Method	R@0.3	R@0.5
LocVTP[5]	41.6	28.9
ProTéGé w/o Untrimmed	41.76	30.01
ProTéGé	**43.63**	**31.19**

Evaluation on VidSitu. To demonstrate the effectiveness of the proposed method on understanding movie data, we further evaluate ProTéGé on VidSitu, which is a large-scale dataset containing diverse videos from movies depicting complex situations. Specifically, we choose the event relation classification as our downstream task and Vid TxEnc as the downstream method. Table 7 shows MacroAveraged Accuracy on validation set. ProTéGé exceeds Vid TxEnc and trimmed pretraining, furthur validating its generalization ability on movie data and complex situation understanding task.

Table 7. Evaluation on VidSitu.

Method	Macro-Acc
Vid TxEnc	34.54
ProTéGé w/o Untrimmed	41.81
ProTéGé	**45.47**

2. Additional Qualitative Analysis

Fig 1 and Fig 2 show the similarity score proposal grid on videos from the Charades-STA and ActivityNet-Captions respectively for ProTéGé and a baseline setup doing pretraining on trimmed videos. Both setups have never seen the videos during pretraining. We feed the videos and text queries through the video and text query encoders respectively and using the output features, obtain the cosine similarity scores for the 2D proposal grid without doing any finetuning on the videos. The cyan dot in the grid in the figures denotes the location of the ground truth proposal for the corresponding query. We can observe that ProTéGé, having been trained on untrimmed videos, can clearly learn to exhibit higher video-text similarity close to the ground truth and lower similarity farther away from the ground truth. But when trained on trimmed videos, there is no visible difference in the similarity scores across proposals. Moreover, the range of similarity scores is also significantly larger for ProTéGé. This shows that our method, pretrained on untrimmed videos, can develop a more fine-grained understanding of the video, leading to more discriminative intra-video features and allowing for more distinguishable video-text similarity across different regions within a video.

Fig 3 and Fig 4 further compare the fully supervised

VTG performance on videos from the Charades-STA and ActivityNet-Captions respectively by visualizing the localization results. We use 2D-TAN [6] as the downstream method and compare ProTéGé with a baseline setup using features from backbone pretrained on trimmed videos. For both datasets, ProTéGé features can ground the text query in the video significantly more precisely while features pretrained on trimmed videos exhibit a large deviation from the ground truth when grounding the query in the video. As shown in Fig 3c, Fig 3d, Fig 4b, and Fig 4c, when the background inside the ground truth location is visually very similar to the outside background, the baseline makes large errors in correctly grounding the query in the video. On the other hand, ProTéGé, due to its ability to learn highly discriminative features within a video via pretraining on untrimmed videos, is able to perform significantly better and provide accurate query localization.

3. Additional Implementation Details

Expanding on the implementation details in the main text, we conduct our experiments using Swin-T [3] pretrained on Kinetics-400 and Swin-B [3] pretrained on Kinetics-600 as the frozen trimmed video encoders, f^{vf}. From Table 1 of the main text, we can see significant improvements on both backbones using ProTéGé that highlights the usefulness of our method across backbones of different sizes. We use Hugging Face's [4] implementation of RoBERTa-base [2] for the frozen text encoder f^{qf}. For downstream VTG tasks, we only use the visual features from our video encoder f^{v} to have a fair comparison with existing methods. For videos longer than 128s, we use a non-overlapping sliding window of 128s for feature extraction. We resize video frames to 224×224 to feed to the video encoder. Before tokenizing the text, we clean it by lower-casing the text, de-accenting the characters, and removing unicode characters, punctuation, and stop words.

References

[1] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021. 2

[2] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 3

[3] 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. 3

[4] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. 3

[5] Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z. Li. ProGCL: Rethinking hard negative mining in graph contrastive learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24332–24346. PMLR, 17–23 Jul 2022. 2

[6] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 3

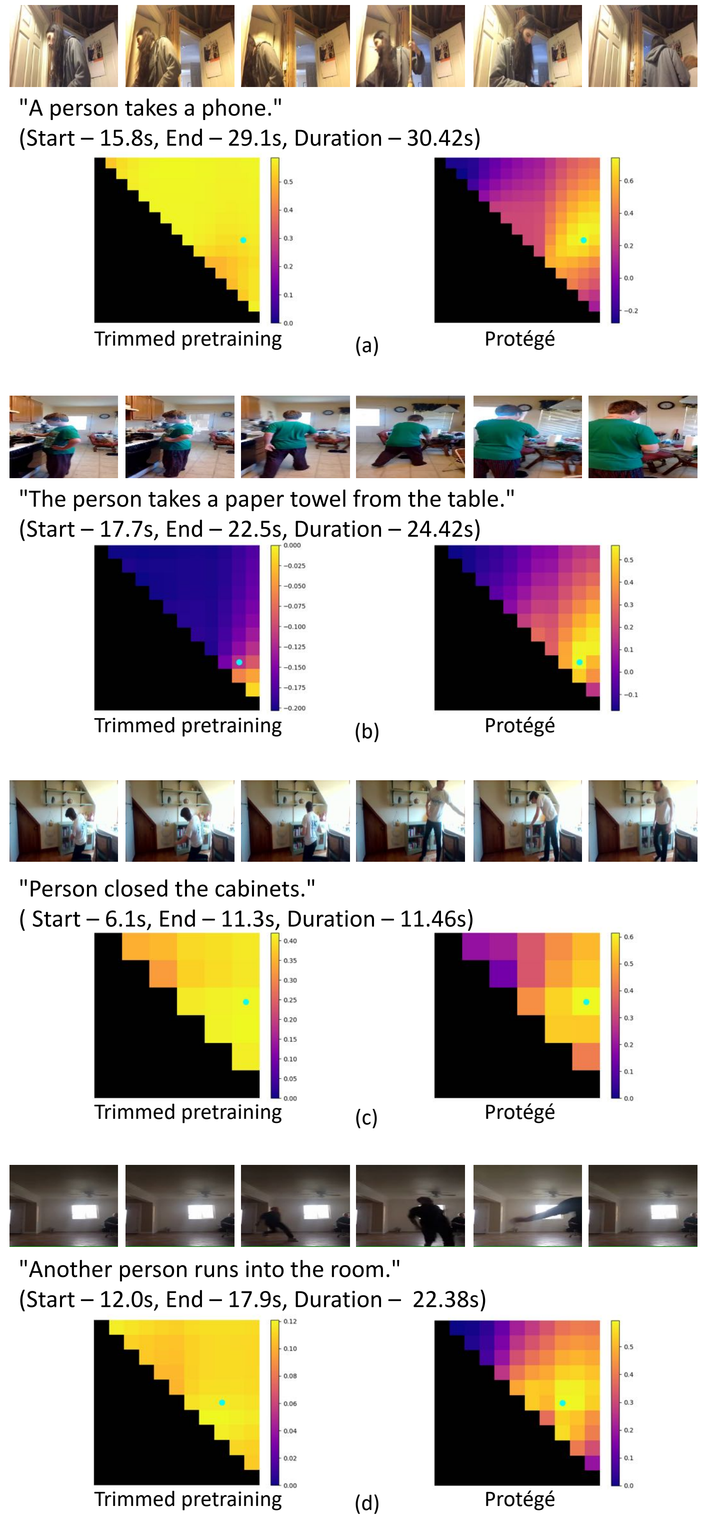

Figure 1. Visualization of 2D proposal grid cosine similarity scores on unseen Charades-STA videos. For each example, the first row shows the video frames and the second row provides the text query along with start-end timestamp and video duration. The third row compares the 2D proposal grid cosine similarity scores of a baseline pretrained on trimmed videos (left) with ProTéGé pretrained on untrimmed videos (right). ProTéGé features show higher variation in cosine similarity with larger similarity closer to the ground truth (cyan dot) due to ProTéGé's ability to learn discriminative features within a video. The grid size varies based on the length of the untrimmed video.

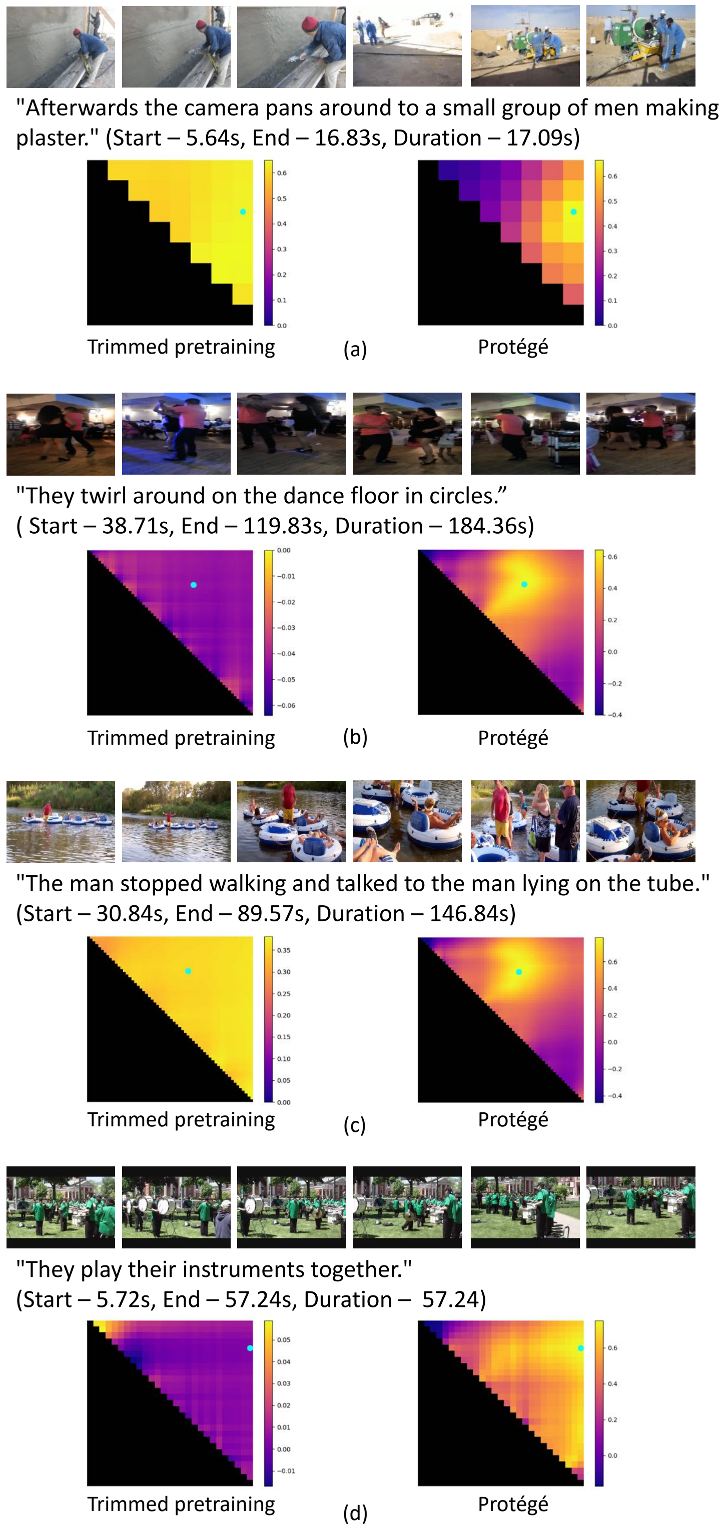

Figure 2. Visualization of 2D proposal grid cosine similarity scores on unseen ActivityNet-Captions videos. For each example, the first row shows the video frames and the second row provides the text query along with start-end timestamp and video duration. The third row compares the 2D proposal grid cosine similarity scores of a baseline pretrained on trimmed videos (left) with ProTéGé pretrained on untrimmed videos (right). ProTéGé features show higher variation in cosine similarity with larger similarity closer to the ground truth (cyan dot) due to ProTéGé's ability to learn discriminative features within a video. The grid size varies as per the length of the untrimmed video.

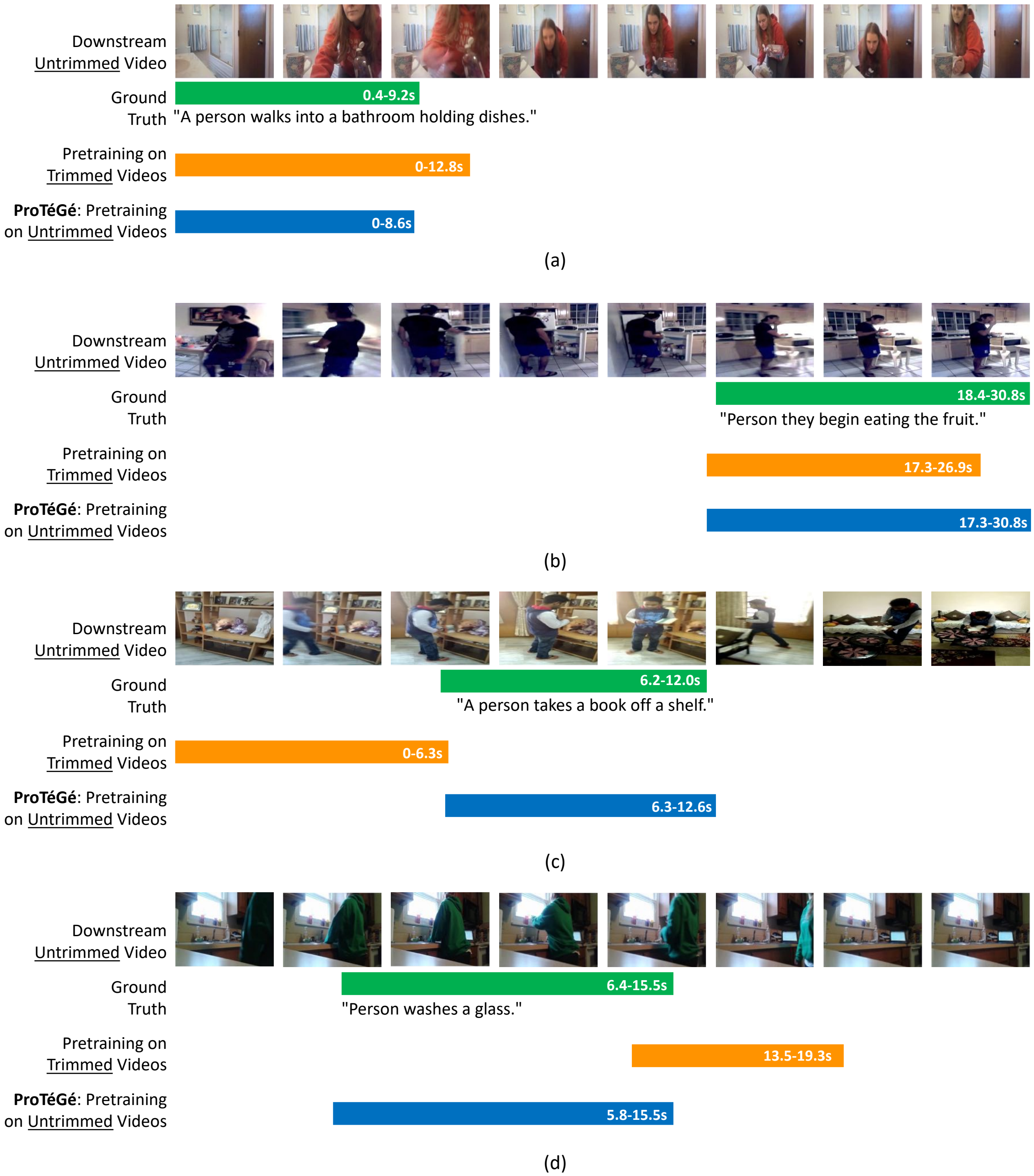

Figure 3. Visualization of fully-supervised video temporal grounding results on Charades-STA dataset using 2D-TAN as the downstream method. For each example, the first row shows the frames of the untrimmed video, the second row shows the ground truth location of the query in the untrimmed video in green, the third row shows grounding prediction in orange from a baseline pretrained on trimmed videos, and the fourth (final) row shows grounding prediction in blue from ProTéGé pretrained on untrimmed videos. We can observe that ProTéGé shows more accurate grounding predictions for all examples as it is pretrained on untrimmed videos which allows ProTéGé to develop a more fine-grained understanding of the video and learn more discriminative features within a video.

Figure 4. Visualization of fully-supervised video temporal grounding results on ActivityNet-Captions dataset using 2D-TAN as the downstream method. For each example, the first row shows the frames of the untrimmed video, the second row shows the ground truth location of the query in the untrimmed video in green, the third row shows grounding prediction in orange from a baseline pretrained on trimmed videos, and the fourth (final) row shows grounding prediction in blue from ProTéGé pretrained on untrimmed videos. We can observe that ProTéGé shows more accurate grounding predictions for all examples as it is pretrained on untrimmed videos which allows ProTéGé to develop a more fine-grained understanding of the video and learn more discriminative features within a video.