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.

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.

<table>
<thead>
<tr>
<th>Untrimmed</th>
<th>Trimmed</th>
<th>Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grounded</td>
<td>Grounded</td>
<td>Grounded</td>
</tr>
</tbody>
</table>

Cosine Similarity of Downstream Untrimmed Video Features with Q1:
- 0.9 vs 0.8
- 13.3 vs 13.4

(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.

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.
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 result, 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 Pretraining for Video Temporal Grounding by Video Temporal 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 algorithm 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 multimodality 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 pretraining, 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
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 \( V = [v_1, \ldots, v_T] \) be an arbitrarily long untrimmed video where \( v_1, \ldots, v_T \) denote the sequence of \( T \) frames forming the video. Let \( Q = [q_1, \ldots, q_L] \) denote a text query comprising a sequence of \( L \) word tokens, \( q_1, \ldots, q_L \). Since ProTeGé 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 ProTeGé. 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
We also randomly sample an arbitrarily long video segment, video frames where each pair is extracted using the subtitles’ start-end times. For the previous works which leverage datasets like HowTo100M\textsuperscript{(1)}\textsuperscript{(vi)}, let \(SB = [SB_1]_{j=1}^n\) be the sequence of \(U\) subtitles. For the \(K\)th sample of our synthesized VTG dataset, we form query \(Q_K\) as \(Q_K = [SB_j]_{j=m}^n = [q_m, \ldots, q_{m+SB_m}, \ldots, q_0, \ldots, q_{n+SB_n}]\), where \(1 \leq m \leq n \leq U\) and \(LSB_m\) and \(LSB_n\) 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 \(V\). We also randomly sample an arbitrarily long video segment, \(V_K = [v_j]_{j=1}^e\) with \(v_s\) and \(v_e\) as start and end frames of \(V_K\) respectively such that \(1 \leq v_s \leq v_e \leq v_{se}\)\textsuperscript{(7)}. We then define the normalized start-end timestamp tuple \((S_K, E_K)\) where \(Q_K\) temporally grounds in \(V_K\) as,

\[
S_K, E_K = \frac{s-s'}{e'-s'}, \frac{e-s'}{e'-s'}, 0 \leq S_K < Q_K \leq 1 \tag{1}
\]

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 \(f^v(\cdot)\) that takes as input an untrimmed video \(V\) with an arbitrary number of frames \(T_V\). We design \(f^v(\cdot)\) to leverage the feature representations learned by the pretrained video encoder \(f_v(\cdot)\) 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 \(f_v(\cdot)\) allows \(f^v(\cdot)\) to focus on learning long-term temporal interactions in an untrimmed video, which is missing in \(f_v(\cdot)\) and is the source of discrepancy between trimmed pretraining and downstream untrimmed VTG. We split arbitrarily long \(V\) into up to \(M\) clips, \(C_V = [C_{V_j}]_{j=1}^M\), 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 \(f^v(\cdot)\) independently and do an average pooling over the frame dimension to obtain a sequence of clip-level local feature representations \(h_V = [h_{V_j}]_{j=1}^M\) of size \(M \times D\) where \(D\) is the feature dimension. \(h_V\) is then fed to a transformer-based untrimmed video encoder, \(f^\text{un}(\cdot)\), to obtain the temporal sequence of feature representations for the untrimmed video as \(z_V = [z_{V_j}]_{j=1}^M\) such that \(z_V = f^v(V) = f^\text{un}(f^v(V))\).

#### Query Encoder.

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

#### Video-Text Similarity-based Grounding Module (VTSGM).

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 (VTSGM) that learns to localize the text query in the untrimmed video. The module first generates an upper triangular 2D proposal grid, \(PG \in \mathbb{R}^{M \times M}\), where each grid cell maps to a video proposal (Fig 2b). Taking output \(z_V = [z_{V_j}]_{j=1}^M\) of video encoder \(f^v(\cdot)\), a cell, \(p_{se}\), at row \(s\) and column \(e\) in proposal grid \(PG\) 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, \(z_{vp_{se}}\), by applying an average pooling over the temporal dimension such that \(z_{vp_{se}} = \frac{\sum_{s \leq \ell \leq e} z_{V_{\ell}}}{e-s+1}\) \(\forall 1 \leq s \leq e \leq M\). Finally, we compute the grid score \(g_{se}\) of each cell \(p_{se}\), as cosine similarity of the text query features \(z_Q\), obtained from the query encoder \(f^q(\cdot)\), with \(z_{vp_{se}}\) for every \(1 \leq s \leq e \leq M\) as,

\[
g(s, e) = \frac{z_{vp_{se}}^T z_Q}{\|z_{vp_{se}}\| \|z_Q\|} \tag{2}
\]

The similarity score \(g_{se}\) reflects how well the region in the untrimmed video represented by \(z_{vp_{se}}\) aligns with the
query text feature \( z_Q \). We take inspiration for the design of \( PG \) from 2D-TAN [51] but we differ significantly in that unlike 2D-TAN, \( PG \) 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, \( L_{loc} \) 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, \( L_{ce} = CrossEntropyLoss(F_V, SE) \), 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, \( PG \) and \( H \) is the total number of video proposals, \( H = \frac{M(M+1)}{2} \) (Fig 2c).

\( L_{ce} \) 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. \( L_{ce} \) alone is not optimal in the presence of noisy ground truth that we have. We, therefore, propose an additional component of \( L_{loc} \) 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, \( D \in \mathbb{R}^{M \times M} \), 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 \( L_{ce} \), we flatten \( D \) to a sequence of distance scores \( D_V \in \mathbb{R}^H \) and compute the distance-based soft-localization loss, \( L_{dist} \) as the L1 loss,

\[
L_{dist} = \left\| F_V, 2 \left( 1 - \frac{D_V - \min(D_V)}{\max(D_V) - \min(D_V)} \right) - 1 \right\|_1
\]

The above formulation affords two benefits, (1) since we use similarity scores to obtain \( z_V \) without involving ground truth, it enables \( L_{inter} \) 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 GumbelSoftmax function keeps the weighted proposal differentiable to allow influencing the features from the entire untrimmed video.

**Intra-Video-Text Alignment Loss** While \( L_{inter} \) 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 \( L_{intra} \) as,
where $z_{V}^{f}$ and $z_{V}^{t}$ 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 2c).

We optimize ProTéGé using a combination of the above three losses defined as $L_{VTG} = L_{loc} + w_{1} \cdot L_{inter} + w_{2} \cdot L_{intra}$ 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, and val 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 $M = 60$. We apply an attention mask for videos shorter than 128s. We use frozen pretrained Video Swin Transformer [25] for trimmed video encoder $f^{v}(.),$ pretrained Roberta [24] for text encoder $f^{t}(.)$, and 4-layer BERT [6] for $f^{vu}(.)$ and $f^{tv}(.)$. 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 $pr = 3$.

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 set 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 1c shows the comparison on ActivityNet-Captions. Comparing ProTéGé
strates the versatility of ProTeGé in improving VTG performance via VTG on downstream VTG tasks. This also demonstrates the effectiveness of our 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 ProTeGé, 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, $L_{loc}$, 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 $L_{CE}$ and $L_{list}$ to perform effective pretraining. Next, we observe that if we instead remove the video-text alignment loss, $L_{inter}$ and $L_{intra}$, 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 $SE$.
duct 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>R@0.5</th>
<th>R@0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>53.26</td>
<td>30.38</td>
</tr>
<tr>
<td>Localization loss</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- w/o $L_{CE}$</td>
<td>51.18</td>
<td>27.33</td>
</tr>
<tr>
<td>- w/o $L_{dist}$</td>
<td>51.86</td>
<td>28.35</td>
</tr>
<tr>
<td>- w/o $L_{CE} + L_{dist}$</td>
<td>49.40</td>
<td>26.60</td>
</tr>
<tr>
<td>Alignment loss</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- w/o $L_{intra}$</td>
<td>52.45</td>
<td>29.22</td>
</tr>
<tr>
<td>- w/o $L_{inter}$</td>
<td>52.39</td>
<td>28.88</td>
</tr>
<tr>
<td>- w/o $L_{intra} + L_{inter}$</td>
<td>51.63</td>
<td>29.59</td>
</tr>
</tbody>
</table>

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, $L_{CE}$, the performance drops by 2.08%/3.05% while removing the distance loss, $L_{dist}$, leads to a reduction of 1.4%/2.03%. This highlights that both $L_{CE}$ and $L_{dist}$ contribute significantly to the pretraining objective. Either soft-localization via $L_{dist}$ or hard-localization via $L_{CE}$ 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, $L_{inter}$ to reduce performance by 0.87%/1.5% and removing intra-alignment loss, $L_{intra}$ 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; $L_{inter}$ leverages it via alignment and $L_{intra}$ uses it to make features within a video more discriminative.

Top-pr proposals in $L_{intra}$. Table 2b further shows the results on selecting a different number of proposals as part of

![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).](image)

Video 1 : “The person pours something into a glass.” (Start -13.3s, End -22.4s, Duration -23.83s)

Video 2 : “One person uses a camera to take a picture.” (Start -17.5s, End -25.8s, Duration -32.71s)

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


[27] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou.


[51] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment local-


