Learning to Focus on the Foreground for Temporal Sentence Grounding

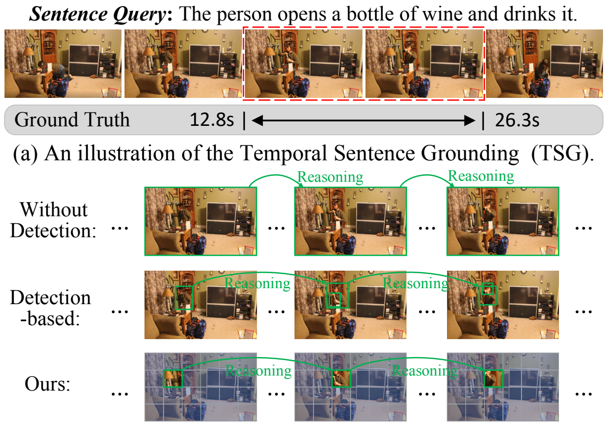
Daizong Liu and **Wei Hu**[*]
Wangxuan Institute of Computer Technology, Peking University
dzliu@stu.pku.edu.cn, forhuwei@pku.edu.cn

Abstract

Temporal sentence grounding (TSG) is crucial and fundamental for video understanding. Previous works typically model the target activity referred to the sentence query in a video by extracting the appearance information from each whole frame. However, these methods fail to distinguish visually similar background noise and capture subtle details of small objects. Although a few recent works additionally adopt a detection model to filter out the background contents and capture local appearances of foreground objects, they rely on the quality of the detection model and suffer from the time-consuming detection process. To this end, we propose a novel detection-free framework for TSG—**G**rounding with **L**earnable **F**oreground (GLF), which efficiently learns to locate the foreground regions related to the query in consecutive frames for better modelling the target activity. Specifically, we first split each video frame into multiple patch candidates of equal size, and reformulate the foreground detection problem as a patch localization task. Then, we develop a self-supervised coarse-to-fine paradigm to learn to locate the most query-relevant patch in each frame and aggregate them among the video for final grounding. Further, we employ a multi-scale patch reasoning strategy to capture more fine-grained foreground information. Extensive experiments on three challenging datasets (Charades-STA, TACoS, ActivityNet) show that the proposed GLF outperforms state-of-the-art methods.

1 Introduction

Temporal sentence grounding (TSG) (Gao et al., 2017; Anne Hendricks et al., 2017) is an important yet challenging topic of video understanding in computer vision. Given an untrimmed video, it aims to retrieve a temporal segment that semantically corresponds to a given sentence query, as shown in Figure 1 (a). Compared to other video-and-language tasks like video captioning (Song et al., 2015; Chu et al., 2015) and video action localization (Shou et al., 2016; Zhao et al., 2017; Xiong et al., 2022), TSG is substantially more challenging as it need not only capture the complicated visual and textual information, but also learn the complex multi-modal interactions among them for modelling the target activity.



(a) An illustration of the Temporal Sentence Grounding (TSG).

(b) Motivation of our proposed method.

Figure 1: (a) An illustrative example of the TSG task. (b) The Illustration of our motivation: we learn to selectively focus on the foreground patch in each frame in a detection-free manner, which alleviates the problems of redundant backgrounds in most previous works (without detection) and the low detection quality in time-consuming detection-based methods. The green box marks the focused region.

To localize the target segment, most previous works either pre-define abundant segment proposals (Chen et al., 2018; Zhang et al., 2019; Yuan et al., 2019; Liu et al., 2021b; Zhang et al., 2020b; Zeng et al., 2020; Mo et al., 2022; Liu et al., 2022a) to match the query semantic for ranking and selection, or employ proposal-free frameworks (Chen et al., 2020; Zhang et al., 2020a; Mun et al., 2020)

[*]Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 5532–5541
October 12–17, 2022.

to directly regress the start/end timestamps of the segment. Although these methods have made significant progress in recent years, they extract the appearance information of each whole frame among the entire video, thus limiting the effective integration of the foreground contexts for modelling the target activity due to the visually similar backgrounds and the missing subtle details of small objects. To alleviate such limitations, a few recent works (Zeng et al., 2021; Liu et al., 2022e) attempt to additionally adopt a pre-processing detection model (*i.e.*, Faster R-CNN (Ren et al., 2015)) that detects the foreground objects for filtering out the background noise. However, they rely on the quality of the detection model while suffering from the time-consuming detection process.

Based on the above considerations, this paper aims to develop a *detection-free* grounding network, which efficiently selects the most query-relevant region in each frame to represent the frame-level features among the entire video for better modelling the target activity. As illustrated in Figure 1(b), in order to effectively represent different local regions in each frame, we divide it into multiple patches that serve as the region candidates to be selected according to their semantic similarity with the query. Once the best patch is determined in each frame during the network learning, they are extracted to model the activity by spatial-temporal correlation reasoning. Compared to previous methods, such detection-free network provides more fine-grained foreground details by filtering out the background regions and capturing the local contexts in an efficient and end-to-end manner, leading to better grounding results.

To this end, we propose a novel TSG model, called **G**rounding with **L**earnable **F**oreground (GLF), which learns to focus on the query-relevant foreground regions among video frames to model the fine-grained target activity for more accurate grounding. Specifically, we reformulate the foreground detection problem as a patch localization task. Considering the spatial-temporal information within the video, we extract 3D spatial-temporal patches instead of 2D spatial ones on the video clips (*i.e.*, several consecutive frames). We first introduce a 3D patch embedding layer to encode the local information of each patch candidate, and concatenate it with an additional global representation extracted from its current video clip. Then, we interact the patch candidates in each clip with the query semantic to learn their matching scores for distinguishing the foreground and background patches. Particularly, we develop a two-level coarse-to-fine paradigm to gradually localize the most relevant (foreground) patch in each clip. At last, we aggregate the representations of the most relevant patches among the entire video to model the target activity. In addition, considering the sizes of the foreground regions may vary in different videos, we further extend the GLF model with multi-scale patch design to capture more fine-grained and complete foreground information for better grounding.

Our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to propose a detection-free grounding framework with the foreground learned for the TSG task. To learn to determine the foreground region in each clip, we split the clip into multiple patch candidates and reformulate the foreground detection problem as a patch localization task.
- We propose a coarse-to-fine self-supervised paradigm to localize the most query-relevant region in each clip for final grounding. We further extend the paradigm with multi-scale patch reasoning in a parallel manner to capture more fine-grained foreground details.
- Comprehensive evaluations on three challenging TSG benchmarks (Charades-STA, TACoS, ActivityNet) demonstrate that our GLF outperforms the state-of-the-art performance.

2 Related Work

Temporal sentence grounding (TSG) is a new task introduced recently (Gao et al., 2017; Anne Hendricks et al., 2017), which aims to localize the most relevant video segment from a video with sentence descriptions. Most previous works (Chen et al., 2018; Zhang et al., 2019; Yuan et al., 2019; Liu et al., 2021b, 2020b, 2021a,c, 2022b; Liu and Hu, 2022) generate multiple segment proposals and then rank them according to the similarity between proposals and the query for selecting the best matching one. Instead of generating complex proposals, some works (Zhang et al., 2020a; Chen et al., 2020; Mun et al., 2020; Liu et al., 2022d,c) directly regress the temporal locations of the target

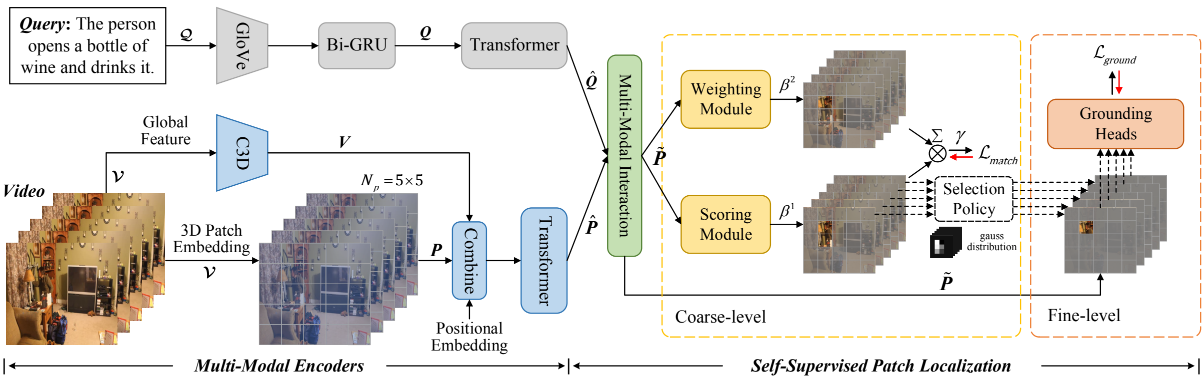


Figure 2: Overview of the proposed GLF model. It consists of the multi-modal encoders and the self-supervised patch localization.

segment by either regressing the start/end timestamps based on the entire video representation or predicting at each frame to determine whether this frame is a start or end boundary. Although the above two types of methods achieve outstanding performance, they all extract the appearance information of each whole frame among the entire video for activity modelling, which fails to capture fine-grained local object details for semantic composing and may suffer from visually similar background contents. A few recent works (Zeng et al., 2021; Liu et al., 2022e) attempt to alleviate such limitations by detecting and learning the correlations between the foreground objects for reasoning the multi-modal semantics. These methods can well filter out the background noise and focus more on local details of small objects. However, they rely on the quality of the time-consuming detection model. In this paper, we propose a detection-free grounding network to learn to focus on the foreground region in each frame for activity composing, which is more efficient than the detection-based methods since our model is trained end-to-end with a learnable foreground attention mechanism. Besides, our model is also more effective than previous proposal-based and proposal-free methods by filtering out the background appearances and capturing more fine-grained subtle details.

3 Our Method

Given an untrimmed video $\boldsymbol{\mathcal{V}}$ and a sentence query $\boldsymbol{\mathcal{Q}}$, the TSG task aims to determine the start and end timestamps of a specific video segment referring to the sentence query. Formally, we represent the video as $\boldsymbol{\mathcal{V}} = \{\boldsymbol{c}_t\}_{t=1}^{T}$ clip-by-clip, where $\boldsymbol{c}_t$ is the t-th clip and T is the total clip number. We also denote the query as $\boldsymbol{\mathcal{Q}} = \{\boldsymbol{w}_s\}_{s=1}^{S}$ word-by-word, where S is the length of the sentence.

In this section, we introduce the overall architecture of our proposed GLF model. As shown in Figure 2, the model consists of two main parts: multi-modal encoders and self-supervised patch localization. First of all, GLF splits each video clip into multiple spatial-temporal patch candidates of equal size and encodes them with a shared 3D embedding layer. Then, the model extracts the query embeddings and interact them with all patch candidates. After that, a coarse-to-fine patch localization paradigm is proposed to gradually score the patches in each clip according to their query-relevant similarity. At last, a single best patch in each clip is selected by a learned policy module to represent the current query-guided clip feature for final grounding. Considering the sizes of the foreground regions may vary in different videos, we further extend the model with a multi-scale patch design to aggregate different-level foreground contexts. We elaborate on each module below.

3.1 Multi-Modal Encoders

Video encoder. For the input video $\boldsymbol{\mathcal{V}}$, to extract different regional local information in the t-th clip $\boldsymbol{c}_t$, we first split $\boldsymbol{c}_t$ into spatial-temporal patch candidates with the same temporal dimension as $\boldsymbol{c}_t$ and no spatial overlap, where the total number of spatial-temporal patches is $N_p = K \times K$ and K is the patch number in each column/row. Then, we take a shared-weight 3D kernel with a further projection layer as the patch embedding module to encode all N_p patches of $\boldsymbol{c}_t$ into $\{\boldsymbol{p}_{t,i}\}_{i=1}^{N_p}$, where $\boldsymbol{p}_{t,i} \in \mathbb{R}^{d_1^v}$ and i denotes the patch index, and d_1^v is the feature dimension. Considering the global feature of the whole clip contains the non-local information across different patches, we also utilize the pre-trained C3D model (Tran et al., 2015) to extract clip-level feature $\boldsymbol{v}_t \in \mathbb{R}^{d_2^v}$ of each clip $\boldsymbol{c}_t$.

Since it is necessary to consider both spatial and temporal locations of each patch for reasoning patch-wise relations, we follow (Dosovitskiy et al., 2021) to encode the spatial position embeddings of each patch $\boldsymbol{p}_{t,i}$ as $\boldsymbol{e}_i^{spa}$, and follow (Mun et al., 2020) to define its temporal position embeddings as $\boldsymbol{e}_t^{tem}$. The final patch-wise feature is concatenated as:

$$(\boldsymbol{p}_{t,i})' = [\boldsymbol{p}_{t,i}; \boldsymbol{v}_t; \boldsymbol{e}_i^{spa}; \boldsymbol{e}_t^{tem}]. \tag{1}$$

We further employ a plain Transformer encoder (Vaswani et al., 2017) on all patches $\{(\boldsymbol{p}_{t,i})'\}_{t=1,i=1}^{t=T,i=N_p}$ to model their intra-modality contexts, and obtain corresponding contextualized representations as $\widehat{\boldsymbol{P}} = \{\widehat{\boldsymbol{p}}_{t,i}\}_{t=1,i=1}^{t=T,i=N_p} \in \mathbb{R}^{T \times N_p \times d_3^v}$.

Query encoder. For the query $\boldsymbol{Q}$, following previous works (Chen et al., 2018; Liu et al., 2021b), we first utilize the Glove model (Pennington et al., 2014) to embed each word into a dense vector, and then employ a Bi-GRU (Chung et al., 2014) to encode its sequential information. The encoded features can be denoted as $\boldsymbol{Q} = \{\boldsymbol{q}_s\}_{s=1}^S \in \mathbb{R}^{S \times d_1^q}$. We also employ another plain Transformer encoder to model the contextual textual representations as $\widehat{\boldsymbol{Q}} = \{\widehat{\boldsymbol{q}}_s\}_{s=1}^S \in \mathbb{R}^{S \times d_2^q}$.

3.2 Self-Supervised Patch Localization

Since there is only temporal-level annotation of the target segment and no spatial-level annotation of the foreground regions, we develop a self-supervised learning paradigm to guide the GLF learn to focus on the potential foreground patches. Specifically, we first interact video and query features to align their relevant semantics, and then impose two cooperated modules on the multi-modal features to compute the clip-query matching scores by scoring and weighting different patches in the same clip. We further propose a coarse-to-fine patch localization strategy to gradually select a patch in each clip to effectively represent the clip-level query-relevant semantic for more fine-grained and accurate grounding.

Multi-modal interaction. To capture the relationship between each patch and the query, we employ a multi-modal interaction module that selectively injects textual evidences into the visual patches. We first utilize an attention mechanism to aggregate the word features for each patch. For the patch $\widehat{\boldsymbol{p}}_{t,i}$, we calculate the attention weights over word features $\{\widehat{\boldsymbol{q}}_s\}_{s=1}^S$ and aggregate them as:

$$\begin{aligned} \alpha_{t,i,s} &= \mathbf{w}^\top \tanh(\mathbf{W}_1^\alpha \widehat{\boldsymbol{p}}_{t,i} + \mathbf{W}_2^\alpha \widehat{\boldsymbol{q}}_s + \mathbf{b}^\alpha), \\ \boldsymbol{r}_{t,i} &= \sum_{s=1}^S \text{softmax}(\alpha_{t,i,s}) \cdot \widehat{\boldsymbol{q}}_s, \end{aligned} \tag{2}$$

where $\mathbf{W}_1^\alpha, \mathbf{W}_2^\alpha$ are projection matrices, $\mathbf{b}^\alpha$ is the bias and $\mathbf{w}^\top$ is the row vector (Zhang et al., 2019). $\boldsymbol{r}_{t,i}$ is the patch-aware textual feature for each patch i in the t-th clip. Next, we build the textual gate that takes language information as the guidance to weaken the text-irrelevant patches, and generate the cross-modal patch features as:

$$\boldsymbol{g}_{t,i} = \sigma(\mathbf{W}^g \boldsymbol{r}_{t,i} + \mathbf{W}^b),\ \widetilde{\boldsymbol{p}}_{t,i} = [\widehat{\boldsymbol{p}}_{t,i} \odot \boldsymbol{g}_{t,i}; \boldsymbol{r}_{t,i}], \tag{3}$$

where σ is the sigmoid function, $\odot$ is the element-wise multiplication, $\boldsymbol{g}_{t,i}$ means the textual gate for patch i. $\widetilde{\boldsymbol{P}} = \{\widetilde{\boldsymbol{p}}_{t,i}\}_{t=1,i=1}^{t=T,i=N_p}$ is the query-guided patch features.

Learning to focus on the foreground. We propose a self-supervised learning paradigm to estimate the potential foreground patches by learning to selectively aggregate the patch information within each clip for clip-query matching. Considering patches of the same clip have different semantic similarities with the query and their contribution to the query-guided clip-level semantic is often quite different, we develop two separate scoring and weighting modules to evaluate the patch-query similarity and patch-to-clip weight, respectively. Both modules are implemented by two linear layers. For the patches in the t-th clip, we formulate the self-supervised learning process as:

$$\begin{aligned} &\beta_{t,i}^1 = \text{scoring}(\widetilde{\boldsymbol{p}}_{t,i}),\ \beta_{t,i}^2 = \text{weighting}(\widetilde{\boldsymbol{p}}_{t,i}), \\ &\gamma_t = \sum_{i=1}^{N_p} \text{softmax}(\beta_{t,i}^2) \cdot \sigma(\beta_{t,i}^1), \end{aligned} \tag{4}$$

where $\beta_{t,i}^1$ represents the score whether the i-th patch in the t-th clip is the query-relevant one, $\beta_{t,i}^2$ is the predicted weight for aggregating all patches within the current clip. γ_t denotes the final clip-query matching score, which represents whether the t-th clip is in the ground-truth segment or not. To prevent the two modules merging into the similar or identical parameters, we utilize the sigmoid function following the scoring module to force it to learn the score whether the patch matches the query, and we utilize the softmax function following the weighting module to predict the weights for aggregating all patches within the current frame. Once the clip-level scores are well-trained, the scoring module can best predict the similarity between each patch and the query, and guide the model to focus more on the foreground patches.

To supervise the above two modules, we use the clips falling into the ground-truth segment as positive samples and the others as negative samples, and formulate a balanced binary cross-entropy loss as:

$$\begin{aligned}\mathcal{L}_{match} = &-\sum_{t=1}^{T_{pos}} \frac{T_{neg}}{T} y_t \log(\gamma_t) \\ &-\sum_{t=1}^{T_{neg}} \frac{T_{pos}}{T} (1-y_t) \log(1-\gamma_t),\end{aligned} \tag{5}$$

where T_{pos}, T_{neg} are the numbers of positive and negative clips. Since most videos are long while the lengths of annotated target segments is short, the numbers of positive and negative clips are unbalance. Therefore, we utilize Tneg/T and Tpos/T to balance their losses. y_t is the ground-truth label that equals to 1 for positive samples and 0 for negative samples.

Coarse-to-fine patch localization. Equation (4) is a vanilla solution to aggregate the potential foreground contexts by patch-wise scoring and weighting. Since the query-related activity mainly appears in one small region of each clip, based on the above coarse foreground localization operation, we further design a fine-level localization module to only select one patch feature to represent its clip-level query-guided semantic for grounding. Specifically, we develop a selection policy module to choose a single patch from a Gaussian distribution which is transformed from the previous predicted patch-wise scores in each clip. There is no learnable parameter for this module, and the patch with a higher patch-wise score will get a larger probability to represent the frame. We denote such generated clip-level representations as $\boldsymbol{F} = \{\boldsymbol{f}_t\}_{t=1}^{T}$, where $\boldsymbol{f}_t$ is the feature of the selected patch in the t-th clip. After that, we apply the effective grounding heads following (Zhang et al., 2019; Liu et al., 2020b,a) on $\boldsymbol{F}$ to generate N_Φ fine-grained segment proposals for ranking via both confidence scoring loss $\mathcal{L}_{iou}$ and boundary adjustment loss $\mathcal{L}_b$ as:

$$\mathcal{L}_{iou} = -\frac{1}{N_\Phi} \sum_{i=1}^{N_\Phi} (IoU_i log(cs_i) + (1-IoU_i) log(1-cs_i)), \tag{6}$$

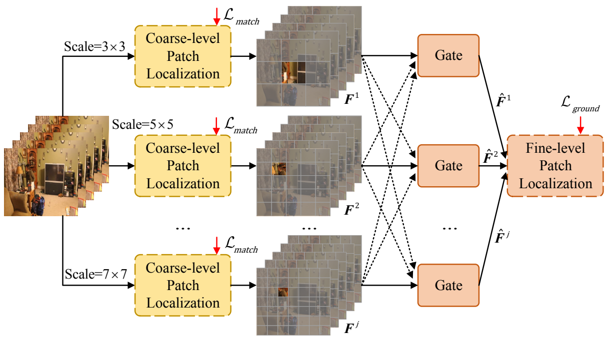


Figure 3: Illustration of the multi-scale patch reasoning strategy.

$$\mathcal{L}_b = \frac{1}{N_{pos}} \sum_{j}^{N_{pos}} \mathcal{R}_1(\hat{\delta}_j^s - \delta_j^s) + \mathcal{R}_1(\hat{\delta}_j^e - \delta_j^e), \tag{7}$$

where cs_i is the predicted confidence score of each segment proposal, $IoU_{t,i}$ is corresponding ground-truth. N_{pos} denotes the number of the positive proposals, and $\mathcal{R}_1$ is the smooth L1 loss. Therefore, the grounding loss can be formulated with a balanced parameter λ as follows:

$$\mathcal{L}_{ground} = \mathcal{L}_{iou} + \lambda \mathcal{L}_b. \tag{8}$$

Let $\phi_{t,i}$ denote the probability of patch $\widetilde{\boldsymbol{p}}_{t,i}$ being selected, the goal of this selection policy module is to minimize $\sum \phi_{t,i} \cdot \mathcal{L}_{ground}$, where $\mathcal{L}_{ground}$ is the reward to enforce it to select the patch that enables the network to produce correct grounding in high confidence. In this manner, the coarse-level localization module gradually finds the important patches of each clip and yields a better clip-wise feature representation. Meanwhile, the selected foreground patch and corresponding representation can further lead to more precise grounding results in the fine-level localization module that in turn provides better supervisions for patch-wise scoring at the coarse level.

3.3 Multi-Scale Patch Aggregation

In order to obtain more reliable foreground information among video clips for final grounding, we exploit the multi-scale property to fuse the contents of the best patch with multiple scales in each clip. Specifically, we split the same video clip into different numbers of patch, and then separately train different patch localization modules for different patch scales. Since a patch with a smaller scale captures major local details and a patch with a larger scale preserves more global contexts, fusing the information from multi-scale best patches in the same clip leads to more representative foreground features. However, directly fusing the multi-scale results cannot take full advantage of the complementary information in different scales. Therefore, we propose a gate-based multi-scale aggregation module to distill each scale patch information for better fusion. Details are illustrated in Figure 3.

For coarse-level patch localization, we extend the multi-scale strategy by learning the self-supervised process in Equation (4) with different patch scales, respectively. For fine-level patch localization, we first separately select one patch in each clip t with multiple scales as $\boldsymbol{f}_t^j$, where j denotes the scale index and $j \in \mathcal{J}$ which is empirically defined. Then, we generate a distilled gate $\boldsymbol{g}_{1,t}^j$ and a reset gate $\boldsymbol{g}_{2,t}^j$ which play a similar role to the gates in LSTM. The gates at each scale control how much the feature at each scale contributes to the final fused feature. This process can be formulated as:

$$(\boldsymbol{f}_t^j)' = (1 - \boldsymbol{g}_{1,t}^j) \odot \boldsymbol{f}_t^j + \sum_{j' \in \mathcal{J}, j' \neq j} \eta^{j',j} \boldsymbol{g}_{1,t}^j \odot \boldsymbol{f}_t^{j'},$$
$$\widehat{\boldsymbol{f}}_t^j = \boldsymbol{g}_{2,t}^j \odot \tanh((\boldsymbol{f}_t^j)') + (1 - \boldsymbol{g}_{2,t}^j) \odot \boldsymbol{f}_t^j, j \in \mathcal{J}, \tag{9}$$

where $\eta^{j',j}$ is a learnable parameter to adjust the relative ratio of the distilled gate which controls information flow of features from a different scale j' combined with the current scale j. $\widehat{\boldsymbol{f}}_t^j$ is the updated patch feature at scale index j, and we concatenate $\{\widehat{\boldsymbol{f}}_t^j\}_{j=1}^{\mathcal{J}}$ of all scales as the fused features $\widetilde{\boldsymbol{f}}_t$ and send it to the grounding heads in Equation (8).

3.4 Training and Testing

Training. To ensure our proposed GLF is trained properly, we propose a three-stage training scheme. At the first stage, we do not integrate the fine-level patch localization module into GLF. Instead, we train the coarse-level one with multi-scale patch definition by minimizing the loss $\mathcal{L}_{match}$ in Equation (5). In this stage, the network is trained to score the foreground patches. At the second stage, we fix the trained network obtained from stage-1, and evoke the fine-level patch localization module with the multi-scale strategy to focus on the selected patch in each clip by minimizing the loss $\mathcal{L}_{ground}$ in Equation (8). At last, we fine-tune the whole GLF model.

Testing. During testing, we select patches of highest scores in the fine-level patch localization module for grounding.

4 Experiments

4.1 Datasets and Evaluation

Charades-STA. This dataset (Gao et al., 2017) consists of 9848 videos of daily life indoor activities. There are 12408 sentence-video pairs for training and 3720 pairs for testing.

TACoS. This dataset (Regneri et al., 2013) collects 127 long videos, which are mainly about cooking scenarios, thus lacking the diversity. We use the same split as [Gao et al., 2017], which has 10146, 4589 and 4083 sentence-video pairs for training, validation, and testing, respectively.

ActivityNet. It is a large dataset (Krishna et al., 2017) which contains 20k videos with 100k language descriptions. This dataset pays attention to more complicated human activities in daily life. Following public split, we use 37417, 17505, and 17031 sentence-video pairs for training, validation, and testing, respectively.

Evaluation. We adopt "R@n, IoU=m" as our evaluation metric, which is defined as the percentage of at least one of top-n selected moments having IoU larger than m.

4.2 Implementation Details

For each video input, we adopt 112×112 pixels of every frame. We define consecutive 16 frames as a clip and each clip overlaps 8 frames with adjacent clips. The kernel size of the 3D patch embedding layer is adaptive to the defined patch size. We extract clip-level global features from a pre-trained C3D (Tran et al., 2015) or I3D (Carreira and Zisserman, 2017) model. Since some videos are overlong, we uniformly downsample clip sequences to $T = 200$ for TACoS, ActivityNet, and $T = 64$ for Charades-STA. For each sentence input, we set the length of word feature sequences to $S = 20$, and utilize Glove (Pennington et al., 2014) to embed each word to 300 dimension features. The hidden dimension of Bi-GRU is 512, and the hyper-parameter λ is set to 0.005. The numbers $N_p = K \times K$ of the split multi-scale patches in each clip are set to 3×3, 5×5, 7×7. We train the whole model with batch size of 64 and early stopping strategy. Parameter optimization is performed by Adam optimizer with leaning rate 4×10^{-4} for Charades-STA and 3×10^{-4} for TACoS, ActivityNet, and linear decay of learning rate and gradient clipping of 1.0.

Method	Charades-STA					TACoS				
	Feature	R@1, IoU=0.5	R@1, IoU=0.7	R@5, IoU=0.5	R@5, IoU=0.7	Feature	R@1, IoU=0.3	R@1, IoU=0.5	R@5, IoU=0.3	R@5, IoU=0.5
CTRL	C3D	23.63	8.89	58.92	29.57	C3D	18.32	13.30	36.69	25.42
QSPN	C3D	35.60	15.80	79.40	45.50	C3D	20.15	15.32	36.72	25.30
CBP	C3D	36.80	18.87	70.94	50.19	C3D	27.31	24.79	43.64	37.40
GDP	C3D	39.47	18.49	-	-	C3D	24.14	-	-	-
VSLNet	I3D	47.31	30.19	-	-	C3D	29.61	24.27	-	-
IVG-DCL	I3D	50.24	32.88	-	-	C3D	38.84	29.07	-	-
DRN	I3D	53.09	31.75	89.06	60.05	C3D	-	23.17	-	33.36
CBLN	I3D	61.13	38.22	90.33	61.69	C3D	38.98	27.65	59.96	46.24
MARN*	C3D+Object	62.08	41.46	91.65	70.03	C3D+Object	43.24	32.70	61.33	51.59
MARN*	I3D+Object	64.31	42.82	93.30	71.76	I3D+Object	45.57	34.06	62.64	52.92
GLF	C3D+Patch	63.60	42.75	92.91	71.49	C3D+Patch	44.82	34.38	62.75	52.26
	I3D+Patch	**65.57**	**44.32**	**94.86**	**73.07**	I3D+Patch	**47.14**	**35.63**	**65.24**	**53.77**

Table 1: Overall performance comparison among our method with proposal-based and proposal-free methods on the Charades-STA and TACoS datasets under the official train/test splits. * denotes that we remove MARN's additional ResNet feature for fair comparison.

Method	Charades-STA					TACoS				
	Feature	R@1, IoU=0.5	R@1, IoU=0.7	R@5, IoU=0.5	R@5, IoU=0.7	Feature	R@1, IoU=0.3	R@1, IoU=0.5	R@5, IoU=0.3	R@5, IoU=0.5
MMRG	Object	44.25	-	60.22	-	Object	57.83	39.28	78.38	56.34
MARN*	C3D+Object	46.19	32.01	63.25	39.88	C3D+Object	59.56	40.47	80.30	58.74
MARN*	I3D+Object	47.67	33.49	65.02	40.51	I3D+Object	61.16	42.33	82.75	59.90
GLF	C3D+Patch	47.85	33.68	64.54	41.20	C3D+Patch	61.37	41.72	81.96	59.45
	I3D+Patch	**49.59**	**35.01**	**66.34**	**42.79**	I3D+Patch	**62.98**	**43.11**	**83.52**	**60.13**

Table 2: Comparison with detection-based method MMRG on Charades-STA and TACoS datasets under MMRG's train/test splits.

Method	ActivityNet				
	Feature	R@1, IoU=0.5	R@1, IoU=0.7	R@5, IoU=0.5	R@5, IoU=0.7
CTRL	C3D	29.01	10.34	59.17	37.54
QSPN	C3D	33.26	13.43	62.39	40.78
CBP	C3D	35.76	17.80	65.89	46.20
GDP	C3D	39.27	-	-	-
VSLNet	C3D	43.22	26.16	-	-
IVG-DCL	C3D	43.84	27.10	-	-
DRN	C3D	45.45	24.36	77.97	50.30
CBLN	C3D	48.12	27.60	79.32	63.41
Ours	C3D+Patch	51.35	30.97	83.26	67.32
	I3D+Patch	**53.48**	**32.15**	**85.02**	**68.81**

Table 3: Overall performance comparison on the ActivityNet dataset under the official train/test splits.

	CTRL	DRN	CBLN	MARN (+detection)	**GLF**
Speed	2.23s	0.15s	0.18s	0.13s (+19.64s)	0.17s

Table 4: Seconds per video on TACoS dataset.

4.3 Comparisons with the State-of-the-art

Compared Methods. To demonstrate the effectiveness of GLF, we compared it with several state-of-the-art methods: Traditional: CTRL (Gao et al., 2017), QSPN (Xu et al., 2019), DRN (Zeng et al., 2020), CBLN (Liu et al., 2021b), CBP (Wang et al., 2020), GDP (Chen et al., 2020), VSLNet (Zhang et al., 2020a), IVG-DCL (Nan et al., 2021); Detection-based: MMRG (Zeng et al., 2021), MARN (Liu et al., 2022e). In particular, the MARN model relies on many types of feature inputs (*i.e.*, C3D, Object, ResNet) for better representation learning. Specifically, their object feature is extracted by detection model, and their ResNet model is utilized to encode such object contexts. Compared to MARN, we only feed single C3D feature as input. Since our method is proposal-free, we re-implement and remove their detector and ResNet models as a new variant MARN* to make a fair comparison with our method.

Comparison on Charades-STA. As shown in Table 1, we reach the highest results over all evaluation metrics on the Charades-STA dataset. Particularly, our C3D+Patch variant outperforms the best detection-based method MARN* by 1.29% and 1.46% in terms of R@1, IoU=0.7 and R@5, IoU=0.7, respectively. Compared to I3D+Patch variant of MARN*, our model also outperforms it by 1.50% and 1.31% in terms of R@1, IoU=0.7 and R@5, IoU=0.7, respectively. We also compare our model following the same data splits of MMRG in Table 2 for fair comparison. It shows that our

Multi-modal Encoders	Self-supervised Patch Localization $\mathcal{L}_{match}$	Self-supervised Patch Localization $\mathcal{L}_{ground}$	Multi-scale Strategy	R@1, IoU=0.7	R@5, IoU=0.7
✓	×	×	×	33.71	62.35
✓	✓	×	×	37.28	65.86
✓	✓	×	✓	40.44	69.17
✓	✓	✓	×	39.53	68.20
✓	✓	✓	✓	**42.75**	**71.49**

Table 5: Main ablation study on Charades-STA dataset.

Components	Variants	R@1, IoU=0.7	R@5, IoU=0.7
Video Encoder	w/o global feature	40.47	70.13
	w/ global feature	**42.75**	**71.49**
	w/o position encoding	39.86	68.94
	w/ position encoding	**42.75**	**71.49**
	w/o transformer	40.38	69.82
	w/ transformer	**42.75**	**71.49**
Query Encoder	w/o Bi-GRU	41.64	70.71
	w/ Bi-GRU	**42.75**	**71.49**
	w/o transformer	41.53	70.70
	w/ transformer	**42.75**	**71.49**

Table 6: Ablation study on the multi-modal encoders.

Components	Variants	R@1, IoU=0.7	R@5, IoU=0.7
Multi-scale Aggregation	w/o gate	41.33	70.16
	w/ gate	**42.75**	**71.49**
Patch Definition	overlap	**42.84**	71.21
	unoverlap	42.75	**71.49**
Scale Size N_p	{9}	40.45	69.27
	{25}	40.72	69.50
	{49}	40.57	69.48
	{81}	40.13	69.06
	{9,25}	42.04	70.79
	{25,49,81}	42.25	70.98
	{9,25,49}	42.75	71.49
	{9,25,49,81}	**42.81**	**71.64**

Table 7: Ablation study on the multi-scale strategy, where scale sizes $9, 25, 49, 81$ denote $3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9$, respectively.

GLF leads to large improvement.

Comparison on TACoS. Table 1 and 2 also show that our GLF achieves the best grounding results on TACoS dataset. Table 1 and 2 also report the grounding results on TACoS dataset. Compared to MARN*, our C3D+Patch model outperforms it by 1.58%, 1.58%, 1.42%, and 0.67% in terms of all metrics. Our I3D+Patch model also outperforms MARN* by a large margin.

Comparison on ActivityNet. Since both MMRG and MARN methods are not implemented on the ActivityNet dataset, we only report the performances on this dataset under official splits as shown in Table 3. Compared to previous best method CBLN, our C3D+Patch model outperforms it by 3.23%, 3.37%, 3.94%, and 3.91% in terms of all metrics. Our I3D+Patch model also outperforms CBLN by a large margin.

Efficiency Comparison. As shown in Table 4, we evaluate the efficiency of our GLF model, by fairly comparing its running time with existing methods on TACoS dataset. It shows that our GLF is more efficient than the detection-based method MARN while on par with the other common methods DRN and CBLN.

4.4 Ablation Study

We perform in-depth ablation studies to evaluate the effectiveness of each component in GLF on Charades-STA dataset. We utilize the C3D+Patch variant as our backbone here.

Main ablation. As shown in Table 5, we verify the contribution of each part in our GLF. We first implement the baseline model by directly applying the grounding heads on the interacted multi-modal features of all patches without both self-supervised patch localization and multi-scale strategy modules. The baseline model achieves 33.71% and 62.35% in terms of R@1, IoU=0.7 and R@5, IoU=0.7, respectively. By adding the coarse-level patch localization module $\mathcal{L}_{match}$ to the baseline, the model brings the improvement of 3.57% and 3.51% since it selectively focuses on the important foreground regions. After further adding the fine-level $\mathcal{L}_{ground}$ for filtering out the redundant patches, the model achieves better results. Besides, the multi-scale strategy also brings a significant improvement to the full model.

Ablation on multi-modal encoders. We also conduct the investigation on different variants of multi-modal encoders in Table 6. We find that the full model performs worse if we remove the global feature that helps to better explore the non-local information among the patches. Besides, it also presents the effectiveness of the position encoding in identifying spatial-temporal knowledge. The transformer modules in both video and query encoders and the Bi-GRU module also bring additional performance to the full model.

Ablation on the multi-scale strategy. We further perform ablation study on our proposed multi-scale patch strategy in Table 7. It shows that our gate-based multi-scale aggregation module brings the improvement of 1.42% and 1.33% in terms of R@1, IoU=0.7 and R@5, IoU=0.7, respectively. Besides, the overlapped and unoverlapped patches have little impact on the final grounding perfor-

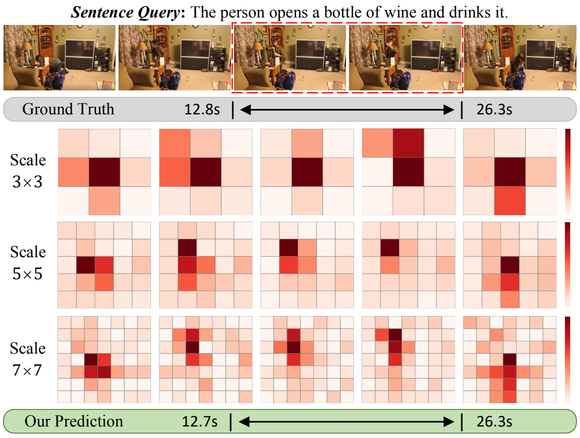


Figure 4: Visualization results on the predicted scores of patches.

mance. Therefore, we choose the unpverlapped one in all our experiments. Moreover, with more various patch scales, the model usually performs better than an individual scale. The variant with four scales $\{9, 25, 49, 81\}$ achieves the best result but only performs marginally better than the three-scale one $\{9, 25, 49\}$ at the expense of a significantly larger cost of GPU memory. Thus, we choose $N_p = \{9, 25, 49\}$ in our all experiments.

4.5 Visualization

We show the visualization on the scored multi-scale patches in Figure 4, where the patches with highest scores contain the most query-related visual appearances. From this figure, we can find that our scoring function can well learn the patch-query similarities among different grains. By jointly combing the contexts from different attended patches, our GLF model performs accurate grounding result.

5 Conclusion

In this paper, we make the first attempt to propose a novel detection-free framework for temporal sentence grounding (TSG), called Grounding with Learnable Foreground (GLF). In particular, we split each video frame into patches with multiple scales, and reformulate the foreground detection problem as a patch localization task. In detail, we interact each patch with the query semantic to learn their matching scores supervised by our newly designed self-supervised losses. Further, we develop a two-level coarse-to-fine paradigm to gradually localize the most query-relevant (foreground) patch in each clip. Moreover, considering the sizes of the foreground regions may vary in different videos, we extend the GLF model with multi-scale patch design to capture more fine-grained and complete foreground information for better grounding. Experimental results on three challenging datasets (Charades-STA, TACoS, ActivityNet) validate the effectiveness of our proposed model.

6 Acknowledgments

This work is supported by the National Key R&D Program of China under contract No. 2021YFF0901502.

References

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *ICCV*.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. 2018. Temporally grounding natural sentence in video. In *EMNLP*.

Long Chen, Chujie Lu, Siliang Tang, Jun Xiao, Dong Zhang, Chilie Tan, and Xiaolin Li. 2020. Rethinking the bottom-up framework for query-based video localization. In *AAAI*.

Wen-Sheng Chu, Yale Song, and Alejandro Jaimes. 2015. Video co-summarization: Video summarization by visual co-occurrence. In *CVPR*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *ICCV*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *ICCV*.

Daizong Liu, Xiang Fang, Wei Hu, and Pan Zhou. 2022a. Exploring optical-flow-guided motion and detection-based appearance for temporal sentence grounding. *arXiv preprint arXiv:2203.02966*.

Daizong Liu and Wei Hu. 2022. Skimming, locating, then perusing: A human-like framework for natural language video localization. In *ACM MM*.

Daizong Liu, Xiaoye Qu, Xing Di, Yu Cheng, Zichuan Xu Xu, and Pan Zhou. 2022b. Memory-guided semantic learning network for temporal sentence grounding. In *AAAI*.

Daizong Liu, Xiaoye Qu, Jianfeng Dong, and Pan Zhou. 2020a. Reasoning step-by-step: Temporal sentence localization in videos via deep rectification-modulation network. In *COLING*.

Daizong Liu, Xiaoye Qu, Jianfeng Dong, and Pan Zhou. 2021a. Adaptive proposal generation network for temporal sentence localization in videos. In *EMNLP*, pages 9292–9301.

Daizong Liu, Xiaoye Qu, Jianfeng Dong, Pan Zhou, Yu Cheng, Wei Wei, Zichuan Xu, and Yulai Xie. 2021b. Context-aware biaffine localizing network for temporal sentence grounding. In *CVPR*.

Daizong Liu, Xiaoye Qu, and Wei Hu. 2022c. Reducing the vision and language bias for temporal sentence grounding. In *ACM MM*.

Daizong Liu, Xiaoye Qu, Xiao-Yang Liu, Jianfeng Dong, Pan Zhou, and Zichuan Xu. 2020b. Jointly cross-and self-modal graph attention network for query-based moment localization. In *ACM MM*.

Daizong Liu, Xiaoye Qu, Yinzhen Wang, Xing Di, Kai Zou, Yu Cheng, Zichuan Xu, and Pan Zhou. 2022d. Unsupervised temporal video grounding with deep semantic clustering. In *AAAI*.

Daizong Liu, Xiaoye Qu, and Pan Zhou. 2021c. Progressively guide to attend: An iterative alignment framework for temporal sentence grounding. In *EMNLP*.

Daizong Liu, Xiaoye Qu, Pan Zhou, and Yang Liu. 2022e. Exploring motion and appearance information for temporal sentence grounding. In *AAAI*.

Shentong Mo, Daizong Liu, and Wei Hu. 2022. Multi-scale self-contrastive learning with hard negative mining for weakly-supervised query-based video grounding. *arXiv preprint arXiv:2203.03838*.

Jonghwan Mun, Minsu Cho, and Bohyung Han. 2020. Local-global video-text interactions for temporal grounding. In *CVPR*.

Guoshun Nan, Rui Qiao, Yao Xiao, Jun Liu, Sicong Leng, Hao Zhang, and Wei Lu. 2021. Interventional video grounding with dual contrastive learning. In *CVPR*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. 2013. Grounding action descriptions in videos. *ACL*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

Zheng Shou, Dongang Wang, and Shih-Fu Chang. 2016. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*.

Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. 2015. Tvsum: Summarizing web videos using titles. In *CVPR*.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Jingwen Wang, Lin Ma, and Wenhao Jiang. 2020. Temporally grounding language queries in videos by contextual boundary-aware prediction. In *AAAI*.

Zeyu Xiong, Daizong Liu, and Pan Zhou. 2022. Gaussian kernel-based cross modal network for spatio-temporal video grounding. *arXiv preprint arXiv:2207.00744*.

Huijuan Xu, Kun He, Bryan A Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. 2019. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*.

Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. 2019. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. In *NIPS*.

Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. 2020. Dense regression network for video grounding. In *CVPR*.

Yawen Zeng, Da Cao, Xiaochi Wei, Meng Liu, Zhou Zhao, and Zheng Qin. 2021. Multi-modal relational graph for cross-modal video moment retrieval. In *CVPR*.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. 2020a. Span-based localizing network for natural language video localization. In *ACL*.

Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. 2020b. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*.

Zhu Zhang, Zhijie Lin, Zhou Zhao, and Zhenxin Xiao. 2019. Cross-modal interaction networks for query-based moment retrieval in videos. In *SIGIR*.

Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. 2017. Temporal action detection with structured segment networks. In *ICCV*.