Gaussian Mixture Proposals with Pull-Push Learning Scheme to Capture Diverse Events for Weakly Supervised Temporal Video Grounding

Sunoh Kim[1], **Jungchan Cho**[2], **Joonsang Yu**[3], **YoungJoon Yoo**[3], **Jin Young Choi**[1]
[1]ASRI, Dept. of Electrical and Computer Eng., Seoul National University
[2]School of Computing, Gachon University,
[3]NAVER CLOVA
{suno8386, jychoi}@snu.ac.kr, thinkai@gachon.ac.kr, {joonsang.yu, youngjoon.yoo}@navercorp.com

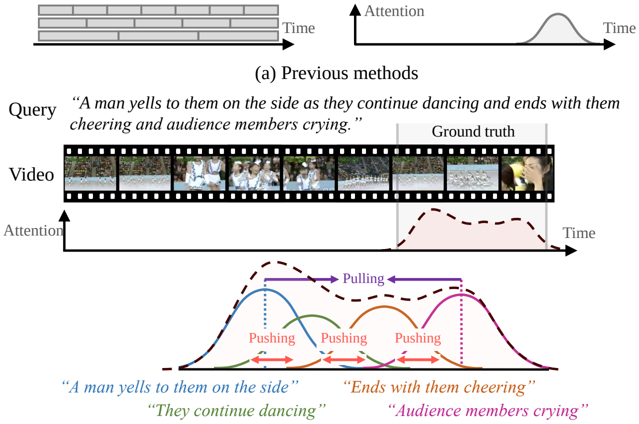
Abstract

In the weakly supervised temporal video grounding study, previous methods use predetermined single Gaussian proposals which lack the ability to express diverse events described by the sentence query. To enhance the expression ability of a proposal, we propose a Gaussian mixture proposal (GMP) that can depict arbitrary shapes by learning importance, centroid, and range of every Gaussian in the mixture. In learning GMP, each Gaussian is not trained in a feature space but is implemented over a temporal location. Thus the conventional feature-based learning for Gaussian mixture model is not valid for our case. In our special setting, to learn moderately coupled Gaussian mixture capturing diverse events, we newly propose a pull-push learning scheme using pulling and pushing losses, each of which plays an opposite role to the other. The effects of components in our scheme are verified in-depth with extensive ablation studies and the overall scheme achieves state-of-the-art performance. Our code is available at https://github.com/sunoh-kim/pps.

1 Introduction

Temporal video grounding is a challenging task in computer vision, where the goal is to find the temporal location of starting and ending points described by a sentence query in an untrimmed video. The task has potential for applications such as video understanding (**?**), video summarization (**?**), and video retrieval (**?**), because it can automatically extract temporal video locations of interest described by given sentences. For temporal video grounding, a fully supervised approach has made remarkable progress (**???**) but require manual annotations of temporal locations for every video-sentence pair. These manual annotations are usually labor-intensive and noisy due to the subjectivity of annotators, which limits their scalability to real-world scenarios and makes trained models biased (**??**).

To overcome the limitation, a weakly supervised approach has been proposed to solve the temporal video grounding problem, where only video-sentence pairs are required for training. Some existing methods (**??????**) use a sliding window strategy to generate proposals for a temporal location but use a lot of pre-defined proposals, which require heavy computation. To reduce the required number of proposals, (**??**) generate learnable Gaussian proposals. However, these single Gaussian proposals with a peak at its center lack the expression ability for diverse query-relevant events in a video.


Time
Attention
Time
(a) Previous methods
Query
"A man yells to them on the side as they continue dancing and ends with them cheering and audience members crying."
Ground truth
Video
Attention
Time
Pulling
Pushing
Pushing
Pushing
"A man yells to them on the side"
"They continue dancing"
"Ends with them cheering"
"Audience members crying"

(b) Proposed methods

Figure 1: Weakly supervised temporal video grounding. (a) Previous methods use sliding windows (left) or a single Gaussian proposal (right), which has a predetermined shape. (b) The proposed method generates a Gaussian mixture proposal trained to be moderately coupled with a pull-push learning scheme to capture diverse query-relevant events.

To enhance the expression ability, we propose a Gaussian mixture proposal (GMP) that can depict arbitrary shapes by learning importance, centroid, and range of every Gaussian in the mixture. Since our GMP is implemented over a temporal location, conventional feature-based learning for Gaussian mixture model (**??**) is not applicable to our approach. In our special setting, our goal is to train the GMP to capture a temporal location semantically relevant to a sentence query that includes diverse events coupled moderately. In Fig. 1, for instance, one sentence query includes two semantic events coupled by "A man yells to them on the side" and "They continue dancing".

To capture the coupled events in a query, we propose a Pull-Push Scheme (PPS) to learn a GMP whose Gaussians are moderately coupled. Specifically, we first define a GMP with learnable parameters: importance, centroid, and range of every Gaussian in the mixture. To learn the importance, we propose an importance weighting strategy that represents importance levels of each Gaussian mask for a query-relevant location. To generate the GMP that represents a query-relevant location, our PPS is trained to reconstruct the sentence query from the proposal. In our scheme, the Gaussians in one GMP should be located near a query-relevant temporal location, but should not be overlapped too much with others to represent diverse events. To this end, our scheme leverages a pulling loss and a pushing loss, each of which plays an opposite role to the other to produce moderately coupled Gaussians. The pulling loss lets the Gaussians stay close to each other by pulling the Gaussian centroids together. The pushing loss prevents the Gaussians from overlapping too much with the others by forcing the Gaussians to be less overlapped.

We verify that our scheme generates high-quality proposals that significantly improve recall rates on the Charades-STA (**?**) and ActivityNet Captions (**?**) datasets. We also demonstrate the effectiveness of each component in our scheme with extensive ablation studies. In summary, our contributions are as follows.

- We generate a Gaussian mixture proposal that represents a query-relevant temporal location by learning importance, centroid, and range of every Gaussian to enhance the expression ability of the proposal.
- We propose a pull-push learning scheme that uses a pulling loss and a pushing loss, each of which plays an opposite role to the other to capture diverse events.
- The proposed components are verified in-depth with extensive ablation studies and the overall scheme achieves state-of-the-art performance.

2 Related Work

Weakly Supervised Temporal Video Grounding

Sliding window-based methods (**?????**) generate proposals through the sliding window strategy and select the most probable proposal. (**?**) proposes a multi-level co-attention model to learn visual-semantic representations. (**?**) uses relations between sentences to understand cross-moment relations in videos. However, sliding window-based methods make a lot of proposals with a predefined length and use Non-Maximum Suppression (NMS) (**?**) to reduce redundant proposals. This process requires a large amount of computation. The proposed method generates learnable Gaussian mixture proposals without using the sliding window.

Reconstruction-based methods (**?????**) assume that well-generated proposals can reconstruct a sentence query from a randomly hidden sentence query. Early works (**??**) aggregate contextual information of video-sentence pairs to score proposals sampled at different scales. However, these methods need to select one proposal from a large set of proposals, which requires heavy computation costs. To solve this problem, other reconstruction-based methods (**??**) propose a learnable Gaussian proposal for a small set of proposals. (**?**) iteratively refines proposal confidence scores to prevent the grounding results from being biased. Unlike the previous methods, our goal is to enhance the expression ability of proposals, hence we generate Gaussian mixture proposals which can effectively represent an arbitrary shape.

Gaussian-based Approach

Gaussians have been studied in various tasks (**??????**). For weakly-supervised temporal video grounding, (**??**) propose learnable Gaussian proposals. Specifically, (**?**) generates one Gaussian proposal for one temporal location, and (**?**) generates multiple Gaussian proposals and selects one proposal to predict a query-relevant temporal location. For action localization, (**?**) uses multiple Gaussian proposals to localize multiple actions, where each single Gaussian proposal represents a temporal location of a specific action.

However, a single Gaussian is a pre-determined shape with a high value at its center, which is not suitable for expressing diverse query-relevant events. To effectively represent the diverse events, we propose a Gaussian mixture proposal by learning importance, centroid, and range of every Gaussian in the mixture. For action localization, (**?**) proposes a layer of Gaussian mixture that replaces a conventional convolutional layer to extract video features. Also, there have been various tasks that train Gaussian mixture model in a feature space (**??**). Unlike these methods, we generate Gaussian mixture proposals that are directly implemented over a temporal location. To represent a query-relevant temporal location that has diverse events, we propose a pull-push scheme to learn the moderately coupled Gaussian mixture.

3 Proposed Method

The overall scheme of the proposed method is depicted in Fig. 2. We generate a new proposal model using multiple learnable Gaussian masks from a video feature $\mathbf{V}$ and a query feature $\mathbf{Q}$. Here, each mask in a video plays a role in focusing on a specific video event and suppressing the rest. We use a mixture model consisting of multiple Gaussian masks to produce proposals. Each positive proposal is called *Gaussian mixture proposal* (GMP). To generate K GMPs ($\mathbf{P}_p$), we propose an importance weighting strategy to represent importance levels of each Gaussian mask for a query-relevant location. For the importance weighting strategy, the importance-based reconstructor receives the generated Gaussian masks and estimates the importance weights of the Gaussian masks for the mixture. Then, the GMP is obtained via attentive pooling with the Gaussian masks and importance weights. To capture diverse query-relevant events, we propose a pull-push learning scheme, where the Gaussian masks are trained by pulling loss and pushing loss. The pulling loss $\mathcal{L}_{pull}$ makes the masks in a GMP be densely overlapped, whereas the pushing loss $\mathcal{L}_{push}$ makes the masks in a GMP be less overlapped. Each of K easy negative proposals ($\mathbf{P}_{en}$) is also composed of multiple Gaussian masks to capture diversely-shaped confusing locations within the given video. Unlike the positive proposal, the easy negative proposal does not use importance weights because the importance weights only represent query-relevant levels, which is only needed for positive proposals. The importance-based reconstructor receives positive proposals

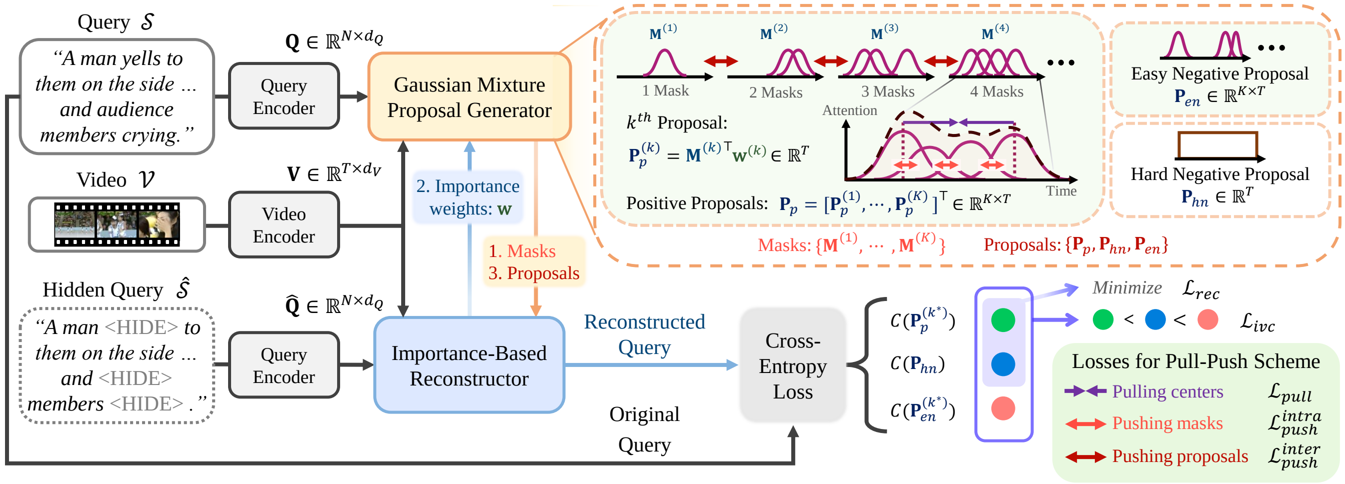

Query $\mathcal{S}$
"A man yells to them on the side ... and audience members crying."
Query Encoder
$\mathbf{Q} \in \mathbb{R}^{N \times d_Q}$
Gaussian Mixture Proposal Generator
Video $\mathcal{V}$
Video Encoder
$\mathbf{V} \in \mathbb{R}^{T \times d_V}$
2. Importance weights: $\mathbf{w}$
1. Masks
3. Proposals
Hidden Query $\hat{\mathcal{S}}$
"A man <HIDE> to them on the side ... and <HIDE> members <HIDE>."
Query Encoder
$\hat{\mathbf{Q}} \in \mathbb{R}^{N \times d_Q}$
Importance-Based Reconstructor
Reconstructed Query
Original Query
Cross-Entropy Loss
$\mathbf{M}^{(1)}$
$\mathbf{M}^{(2)}$
$\mathbf{M}^{(3)}$
$\mathbf{M}^{(4)}$
1 Mask
2 Masks
3 Masks
4 Masks
k^{th} Proposal:
$\mathbf{P}_p^{(k)} = \mathbf{M}^{(k)\top}\mathbf{w}^{(k)} \in \mathbb{R}^T$
Attention
Time
Positive Proposals: $\mathbf{P}_p = [\mathbf{P}_p^{(1)}, \cdots, \mathbf{P}_p^{(K)}]^\top \in \mathbb{R}^{K \times T}$
Easy Negative Proposal
$\mathbf{P}_{en} \in \mathbb{R}^{K \times T}$
Hard Negative Proposal
$\mathbf{P}_{hn} \in \mathbb{R}^T$
Masks: $\{\mathbf{M}^{(1)}, \cdots, \mathbf{M}^{(K)}\}$
Proposals: $\{\mathbf{P}_p, \mathbf{P}_{hn}, \mathbf{P}_{en}\}$
$C(\mathbf{P}_p^{(k^*)})$
$C(\mathbf{P}_{hn})$
$C(\mathbf{P}_{en}^{(k^*)})$
Minimize $\mathcal{L}_{rec}$
● < ● < ● $\mathcal{L}_{ivc}$
Losses for Pull-Push Scheme
Pulling centers $\mathcal{L}_{pull}$
Pushing masks $\mathcal{L}_{push}^{intra}$
Pushing proposals $\mathcal{L}_{push}^{inter}$

Figure 2: The overall scheme of the proposed method. The Gaussian mixture proposal generator produces multiple Gaussian masks from the features representing both the video and sentence query. For the positive proposals, we define a Gaussian mixture proposal, where multiple Gaussian masks are combined via attentive pooling using the importance weights from the importance-based reconstructor. Further, to generate moderately coupled masks in the mixture proposal, we propose the pull-push learning scheme using $\mathcal{L}_{pull}$, $\mathcal{L}_{push}^{intra}$, and $\mathcal{L}_{push}^{inter}$. The importance-based reconstructor leverages the proposals to produce the reconstructed query from the hidden query.

from the Gaussian mixture proposal generator and reconstructs the sentence query from a randomly hidden sentence query.

Encoders

Given a video and sentence query, we use pre-trained encoders to obtain a video feature and a query feature, following previous methods (**??**)

Video encoder. An untrimmed raw video $\mathcal{V}$ is made into a video feature $\mathbf{V}$ through the pre-trained 3D Convolutional Neural Network (3D CNN) (**??**). The video feature $\mathbf{V}$ is given by $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T]^\top \in \mathbb{R}^{T \times d_V}$, where $\mathbf{v}_t$ is the t^{th} segment feature, T is the number of video segments, and d_V is the dimension of the segment feature $\mathbf{v}_t$.

Query encoder. Given a sentence query $\mathcal{S}$, we use the pre-trained GloVe (**?**) word embedding to obtain a query feature $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N]^\top \in \mathbb{R}^{N \times d_Q}$, where $\mathbf{q}_n$ is the n^{th} word feature, N is the number of words, and d_Q is the dimension of the word feature $\mathbf{q}_n$.

Gaussian Mixture Proposal Generator

From video and query features $\mathbf{V}$ and $\mathbf{Q}$, the proposed GMP generator yields K positive GMPs ($\mathbf{P}_p$), K easy negative proposals ($\mathbf{P}_{en}$) in addition to one existing hard negative proposal ($\mathbf{P}_{hn}$).

Modeling of GMP for positive proposal. For the generation of the positive proposal, we first extract a multi-modal feature $\mathbf{G}$ reflecting both visual and textual information. We use a transformer (**?**) to aggregate the information of $\mathbf{V}$ and $\mathbf{Q}$ by $\mathbf{G} = f_{td}(\widehat{\mathbf{V}}, f_{te}(\mathbf{Q})) = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_T, \mathbf{g}_{cls}]^\top \in \mathbb{R}^{(T+1) \times d_G}$, where the transformer uses $\mathbf{Q}$ as an input to the transformer encoder $f_{te}(\cdot)$ and both $\widehat{\mathbf{V}}$ and $f_{te}(\mathbf{Q})$ as inputs to the transformer decoder $f_{td}(\cdot)$, and d_G is the dimension of the multi-modal feature. For the video feature $\widehat{\mathbf{V}}$, we append a learnable token $\mathbf{v}_{cls}$, same as a [CLASS] token in (**?**), by $\widehat{\mathbf{V}} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T, \mathbf{v}_{cls}]^\top \in \mathbb{R}^{(T+1) \times d_V}$. By the transformer, correspondingly, the vector $\mathbf{g}_{cls} \in \mathbb{R}^{d_G}$ stores the sequence information of all words and video segments.

For the k^{th} positive proposal $\mathbf{P}_p^{(k)}$, we define multiple Gaussian masks $\mathbf{M}^{(k)} = [\mathbf{M}_1^{(k)}, \mathbf{M}_2^{(k)}, \ldots, \mathbf{M}_{E_p}^{(k)}]^\top \in \mathbb{R}^{E_p \times T}$, where $\mathbf{M}_l^{(k)}$ is the l^{th} Gaussian mask for $\mathbf{M}^{(k)}$, and E_p is the number of masks. The k^{th} proposal $\mathbf{P}_p^{(k)}$ is defined by a mixture of the Gaussian masks $\mathbf{M}^{(k)}$. The Gaussian centers $\mathbf{c}^{(k)}$ and widths (standard deviations) $\mathbf{s}^{(k)}$ of $\mathbf{M}^{(k)}$ are calculated by the function of $\mathbf{g}_{cls}$, as

$$\mathbf{c}^{(k)} = \text{Sigmoid}\left(\mathbf{W}_\mathbf{c}\, \mathbf{g}_{cls} + \mathbf{b}_\mathbf{c}\right) \in \mathbb{R}^{E_p}, \qquad (1)$$

$$\mathbf{s}^{(k)} = \frac{1}{\sigma}\text{Sigmoid}\left(\mathbf{W}_\mathbf{s}\, \mathbf{g}_{cls} + \mathbf{b}_\mathbf{s}\right) \in \mathbb{R}^{E_p}. \qquad (2)$$

Here, $\mathbf{W}_{\mathbf{c}\text{ or }\mathbf{s}}$ and $\mathbf{b}_{\mathbf{c}\text{ or }\mathbf{s}}$ are defined as learnable parameters of a fully connected layer, and σ is a hyper-parameter controlling the width of the masks. Consequently, we obtain the l^{th} Gaussian mask $\mathbf{M}_l^{(k)} = [f_l^{(k)}(0), f_l^{(k)}(1), \ldots, f_l^{(k)}(T-1)]^\top \in \mathbb{R}^T$ using

$$f_l^{(k)}(t) = \exp\left(-\left(\frac{t/(T-1) - \mathbf{c}_l^{(k)}}{\mathbf{s}_l^{(k)}}\right)^2\right), \qquad (3)$$

where $\mathbf{c}_l^{(k)}$, $\mathbf{s}_l^{(k)} \in \mathbb{R}$ are the l^{th} elements of $\mathbf{c}^{(k)}$, $\mathbf{s}^{(k)}$, respectively.

The k^{th} proposal $\mathbf{P}_p^{(k)}$ is defined by a mixture of the Gaussian masks $\mathbf{M}^{(k)}$ via attentive pooling with mask importance weights $\mathbf{w}^{(k)} \in \mathbb{R}^{E_p}$. Finally, we generate K positive proposals $\mathbf{P}_p = [\mathbf{P}_p^{(1)}, \mathbf{P}_p^{(2)}, \ldots, \mathbf{P}_p^{(K)}]^\top \in \mathbb{R}^{K \times T}$, where the k^{th} proposal is

$$\mathbf{P}_p^{(k)} = \mathbf{M}^{(k)\top}\mathbf{w}^{(k)} \in \mathbb{R}^T. \qquad (4)$$

To represent the importance levels of each Gaussian mask in the mixture, we leverage an importance weighting strategy, where the importance weights $\mathbf{w}^{(k)}$ are estimated by the importance-based reconstructor in Eq. (10).

Losses for pull-push learning scheme. In our scheme, the Gaussian masks in a Gaussian mixture proposal should be densely located near a query-relevant temporal location, but should not be overlapped too much with each other to represent diverse events. To this end, we propose a pull-push learning scheme using a pulling loss and a pushing loss, each of which plays an opposite role to the other, to produce moderately coupled masks.

The pulling loss $\mathcal{L}_{pull}$ lets the masks stay close, which is computed by minimizing the Euclidean distance between the centers of the two farthest masks as follows:

$$\mathcal{L}_{pull} = \sum_{k=1}^{K} \left(\mathbf{c}_{l_{min}}^{(k)} - \mathbf{c}_{l_{max}}^{(k)}\right)^2, \tag{5}$$

where $l_{min} = \arg\min_l \mathbf{c}_l^{(k)}$ and $l_{max} = \arg\max_l \mathbf{c}_l^{(k)}$.

The pushing loss is defined by two losses: (1) an intra-pushing loss and (2) an inter-pushing loss. The intra-pushing loss $\mathcal{L}_{push}^{intra}$ prevents the masks in a proposal from overlapping too much with others by forcing the masks to be less overlapped, which ensures each mask represents different events. Furthermore, we use the inter-pushing loss $\mathcal{L}_{push}^{inter}$ to let each proposal predict different temporal locations. Based on the regularization term in (**?**), the resultant two pushing losses are given as

$$\mathcal{L}_{push}^{intra} = \sum_{k=1}^{K} ||\mathbf{M}^{(k)}\mathbf{M}^{(k)\top} - \lambda_1 I||_F^2, \tag{6}$$

$$\mathcal{L}_{push}^{inter} = ||\mathbf{P}_p\mathbf{P}_p^\top - \lambda_2 I||_F^2, \tag{7}$$

where $|| \cdot ||_F$ denotes the Frobenius norm, I is an identity matrix, and λ_1 and λ_2 are hyper-parameters controlling the strength of the pushing.

Negative proposal mining. To capture diverse shapes of confusing temporal locations inside the video, we generate a new type of a negative proposal with multiple Gaussian masks, called easy negative proposals ($\mathbf{P}_{en} \in \mathbb{R}^{K \times T}$) in addition to the existing hard negative proposal ($\mathbf{P}_{hn} \in \mathbb{R}^T$). To generate K easy negative proposals, we leverage multiple Gaussian masks to include confusing locations. In our negative proposal mining, the k^{th} easy negative proposal ($\mathbf{P}_{en}^{(k)}$) is composed of multiple Gaussian masks by using the same process in Eqs. (1) to (3). Contrary to moderately coupled Gaussian masks in the positive proposal, we let the E_{en} Gaussian masks of each easy negative proposal spread sparsely without the pull-push learning scheme because most of the confusing locations exist throughout the entire video. Then, following (**??**), the hard negative proposal $\mathbf{P}_{hn}$ is determined by a mask covering an entire video, which is $\mathbf{P}_{hn} = [1, 1, \ldots, 1] \in \mathbb{R}^T$, where both the query-relevant location and confusing locations are included. Finally, the Gaussian mixture proposal generator produces three proposals $\{\mathbf{P}_p, \mathbf{P}_{hn}, \mathbf{P}_{en}\}$.

Importance-based Reconstructor

We propose an importance weighting strategy to effectively represent importance levels of each Gaussian mask in the mixture. The importance-based reconstructor produces mask importance weights ($\mathbf{w}$) for Gaussian mixture proposals in Eq. (4). Moreover, the reconstructor receives proposals from the generator and reconstructs the sentence query.

Mask importance. We estimate the k^{th} mask importance weights ($\mathbf{w}^{(k)}$) from the Gaussian masks $\mathbf{M}^{(k)}$. First, we use a Mask-Conditioned transformer (MC transformer) (**??**) to extract the multi-modal feature $\mathbf{R}^{\mathbf{M}}$ for any video mask $\mathbf{M}$, given the video feature $\mathbf{V}$ and a randomly hidden sentence query feature $\widehat{\mathbf{Q}}$. In the MC transformer, the mask $\mathbf{M}$ is multiplied by the self-attention map in every self-attention process to focus on the video feature inside the mask. Additionally, we append a learnable token $\widehat{\mathbf{q}}_{cls}$, same as a [CLASS] token in (**?**), to the hidden sentence query feature by $\widehat{\mathbf{Q}} = [\widehat{\mathbf{q}}_1, \widehat{\mathbf{q}}_2, \ldots, \widehat{\mathbf{q}}_N, \widehat{\mathbf{q}}_{cls}]^\top \in \mathbb{R}^{(N+1) \times d_Q}$. The resultant multi-modal feature $\mathbf{R}^{\mathbf{M}}$ can be calculated as follows:

$$\mathbf{R}^{\mathbf{M}} = f_{md}(\widehat{\mathbf{Q}}, f_{me}(\mathbf{V}, \mathbf{M}), \mathbf{M}) \in \mathbb{R}^{(N+1) \times d_R}. \tag{8}$$

Here, the MC transformer uses $\mathbf{V}$ and $\mathbf{M}$ as inputs to the transformer encoder ($f_{me}(\cdot)$). Then, the transformer decoder ($f_{md}(\cdot)$) receives $\widehat{\mathbf{Q}}$, $f_{me}(\mathbf{V}, \mathbf{M})$, and $\mathbf{M}$. The dimension of the multi-modal feature is denoted by d_R. In $\mathbf{R}^{\mathbf{M}} = [\mathbf{r}_1^{\mathbf{M}}, \mathbf{r}_2^{\mathbf{M}}, \ldots, \mathbf{r}_N^{\mathbf{M}}, \mathbf{r}_{cls}^{\mathbf{M}}]^\top$, the vector $\mathbf{r}_{cls}^{\mathbf{M}}$ reflects all words and video segments conditioned by the mask $\mathbf{M}$. To compute the k^{th} mask importance weights $\mathbf{w}^{(k)}$ in Eq. (4), we calculate $\mathbf{r}_{cls}^{\mathbf{M}_l^{(k)}}$ using $\mathbf{M}_l^{(k)}$ via Eq. (8) and apply it to a Multi-Layer Perceptron (MLP) with two layers as follows:

$$h_l^{(k)} = \mathrm{MLP}\Big(\mathbf{r}_{cls}^{\mathbf{M}_l^{(k)}}\Big) \in \mathbb{R}, \tag{9}$$

$$\mathbf{w}^{(k)} = \mathrm{Softmax}\Big([h_1^{(k)}, h_2^{(k)}, \ldots, h_{E_p}^{(k)}]^\top\Big) \in \mathbb{R}^{E_p}. \tag{10}$$

Losses for reconstruction. Based on the supposition that properly generated proposals can reconstruct the given sentence query as in (**??**), we reconstruct the sentence query from a randomly hidden sentence query. First, we generate the multi-modal features $\mathbf{R}^{\mathbf{P}}$ using the proposed proposals $\mathbf{P} \in \{\mathbf{P}_p, \mathbf{P}_{hn}, \mathbf{P}_{en}\}$ by replacing $\mathbf{M}$ with $\mathbf{P}$ in Eq. (8). Then, the reconstructed query is produced using $\mathbf{R}^{\mathbf{P}}$, and the cross-entropy loss $C(\cdot)$ is used to measure the difference between the reconstructed query and the original query. Then, we can calculate $C(\mathbf{P}_p^{(k)})$, $C(\mathbf{P}_{hn})$, and $C(\mathbf{P}_{en}^{(k)})$. For learning to reconstruct the sentence query, following (**?**), we use a reconstruction loss which is the cross-entropy losses of the positive proposals and hard negative proposal, where a query-relevant temporal location exists, as $\mathcal{L}_{rec} = C(\mathbf{P}_p^{(k^*)}) + C\left(\mathbf{P}_{hn}\right)$, where $k^* = \arg\min_k C(\mathbf{P}_p^{(k)})$. Furthermore, following (**?**), we perform contrastive learning to distinguish the query-relevant location from the confusing locations captured by the easy negative proposals and the hard negative proposal. Based on the triplet loss (**?**), the intra-video contrastive loss $\mathcal{L}_{ivc}$ is defined as $\mathcal{L}_{ivc} = \max\big(C(\mathbf{P}_p^{(k^*)}) - C(\mathbf{P}_{hn}) + \beta_1, 0\big) + \max\big(C(\mathbf{P}_p^{(k^*)}) - C(\mathbf{P}_{en}^{(k^*)}) + \beta_2, 0\big)$, where β_1 and β_2 are hyper-parameters for margins and $\beta_1 < \beta_2$.

Method	R@1,IoU=			R@5,IoU=		
	0.1	0.3	0.5	0.1	0.3	0.5
Random	38.23	18.64	7.63	75.74	52.78	29.49
WS-DEC	62.71	41.98	23.34	-	-	-
MARN	-	47.01	29.95	-	72.02	57.49
VCA	67.96	50.45	31.00	92.14	71.79	53.83
EC-SL	68.48	44.29	24.16	-	-	-
SCN	71.48	47.23	29.22	90.88	71.56	55.69
RTBPN	73.73	49.77	29.63	93.89	79.89	60.56
LCNet	78.58	48.49	26.33	93.95	82.51	62.66
CCL	-	50.12	31.07	-	77.36	61.29
WSTAN	79.78	52.45	30.01	93.15	79.38	63.42
FSAN	78.45	55.11	29.43	92.59	76.79	63.32
CWSTG	71.86	46.62	29.52	93.75	80.92	66.61
CPL	<u>82.55</u>	55.73	31.37	87.24	63.05	43.13
CRM*	81.61	55.26	32.19	-	-	-
CNM*	78.13	55.68	<u>33.33</u>	-	-	-
IRON*	**84.42**	<u>58.95</u>	**36.27**	**96.74**	**85.60**	<u>68.52</u>
PPS	81.84	**59.29**	31.25	<u>95.28</u>	<u>85.54</u>	**71.32**

Table 1: Performance comparisons on the ActivityNet Captions. The best results and second best results are represented as bold and underlined numbers, respectively. The methods using additional annotations or large-scale pre-trained models are marked with *.

Training and Inference

Training. In an end-to-end manner, we train our network with five loss terms: 1) reconstruction loss $\mathcal{L}_{rec}$, 2) intra-video contrastive loss $\mathcal{L}_{ivc}$, 3) pulling loss $\mathcal{L}_{pull}$, and two pushing losses of 4) intra-pushing loss $\mathcal{L}^{intra}_{push}$ and 5) inter-pushing loss $\mathcal{L}^{inter}_{push}$. Then the total loss is given by $\mathcal{L}_{total} = \mathcal{L}_{rec} + \alpha_1\mathcal{L}_{ivc} + \alpha_2\mathcal{L}_{pull} + \alpha_3\mathcal{L}^{intra}_{push} + \alpha_4\mathcal{L}^{inter}_{push}$, where α_1, α_2, α_3, and α_4 are hyper-parameters to balance losses.

Inference. To select the top-1 proposal from the K positive proposals, we use vote-based selection to choose the best overlapping proposal, similar to (**??**).

4 Experiments

Experimental Setup

Evaluation metrics. Following the evaluation metrics in (**?**), we adopt two metrics ('R@n,IoU=m' and 'R@n,mIoU'). 'R@n,IoU=m' denotes the percentage of at least one of the top-n predicted temporal locations having a temporal Intersection over Union (IoU) with a ground truth larger than m. 'R@n,mIoU' denotes the average of the highest IoUs among the n predicted temporal locations.

The ActivityNet Captions dataset (**?**) contains 37,417, 17,505, and 17,031 video-sentence pairs for training, validating val_1, and val_2, respectively. Since a testing set is not publicly available, val_2 is used for testing. Video segment features are extracted via C3D (**?**). Vocabulary sizes are 8,000. For proposals, K, E_{en}, and σ are set to 5, 2, and 4. For losses, α_1, α_2, α_3, and α_4 are set to 1, 0.2, 0.01, and 0.1.

The Charades-STA dataset (**?**) contains 16,128 video-sentence pairs from 6,672 videos, which are divided into 12,408 for training and 3,720 for testing. Video segment features are extracted via I3D (**?**). Vocabulary sizes are 1,111. For proposals, K, E_{en}, and σ are set to 7, 3, and 9. For losses, α_1, α_2, α_3, and α_4 are set to 3, 5, 0.001, and 1.

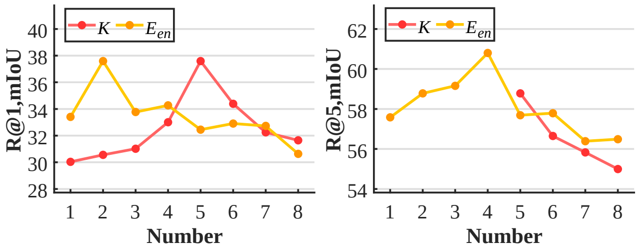


Figure 3: Ablation studies by varying the number of positive and negative proposals K and the number of Gaussian masks of an easy negative proposal E_{en}.

Implementation details. We set the maximum number of video segments to 200, and the maximum length of the sentence query to 20. For the transformers, we use transformers with three-layer and four attention heads. The dimension of the features (d_V, d_Q, d_G, d_R) is set to 256. We use the equivalent MC Transformer for every reconstruction process. For the hidden sentence query, we randomly hide a third (1/3) of the words. For training, the Adam optimizer (**?**) is used. We set the learning rate to 0.0004, mini-batch size to 32, and hyper-parameters as $\lambda_1 = \lambda_2 = 0.15$, $\beta_1 = 0.1$, and $\beta_2 = 0.15$. In the k^{th} positive proposal, we set the number of Gaussian masks E_p to k for reflecting a varying number of masks in each proposal, as shown in the top right of Fig. 2.

Comparison with State-of-the-Art Methods

To verify the effectiveness of the proposed method, we compare our PPS with previous weakly supervised temporal video grounding methods: WS-DEC (**?**), TGA (**?**), SCN (**?**), WSTAN (**?**), VLANet (**?**), MARN (**?**), CCL (**?**), RTBPN (**?**), EC-SL (**?**), LoGAN (**?**), VCA (**?**), LCNet (**?**), FSAN (**?**), CWSTG (**?**), CPL (**?**), CRM (**?**), CNM (**?**), and IRON (**?**).

In Tab. 1 for the ActivityNet Captions dataset, our PPS outperforms CPL (**?**) by 3.56%, 22.49%, and 28.19% at R@1,IoU=0.3, R@5,IoU=0.3, and R@5,IoU=0.5, respectively. It is worth noting that PPS outperforms the previous learnable mask-based method, CPL, by significant margins at R@5, which means that the generated proposals of PPS promise a higher level of quality. In Tab. 3 for the Charades-STA dataset, our PPS surpasses CPL (**?**) by 3.77% and 2.19% at R@1,IoU=0.7 and R@5,IoU=0.3, respectively. The methods marked with * make unfair comparisons with the previous methods. CRM (**?**) uses additional paragraph description annotations. CNM (**?**) uses CLIP large-scale pre-trained features (**?**) and IRON (**?**) uses OATrans (**?**) and DistilBERT (**?**) large-scale pre-trained features. Although our PPS uses 3D ConvNet and Glove features for fair comparisons with previous methods, PPS shows competitive or higher performance with the methods marked with *.

Component	Strategy	Loss	R@1		R@5	
		$\mathcal{L}_{pull}$ & $\mathcal{L}_{push}^{intra}$	IoU=0.3	mIoU	IoU=0.3	mIoU
Proposal type	Single Gaussian	✗	47.49	33.33	78.23	54.85
	Gaussian mixture	✓	**59.29**	**37.59**	**85.54**	**58.78**
Gaussian generation	Learning one center & multiple widths	✗	46.63	31.54	83.65	**59.49**
	Learning multiple centers & widths	✓	47.82	32.08	84.09	56.35
	Learning multiple centers & one width	✓	**59.29**	**37.59**	**85.54**	58.78
Importance weighting	No importance	✓	52.55	34.56	79.67	58.41
	Importance from the generator	✓	48.55	32.36	78.96	56.87
	Importance from the reconstructor	✓	**59.29**	**37.59**	**85.54**	**58.78**

Table 2: Ablation studies of Gaussian mixture proposals on the ActivityNet Captions dataset.

Method	R@1,IoU=			R@5,IoU=		
	0.3	0.5	0.7	0.3	0.5	0.7
Random	20.12	8.61	3.39	68.42	37.57	14.98
TGA	32.14	19.94	8.84	86.58	65.52	33.51
SCN	42.96	23.58	9.97	95.56	71.80	38.87
WSTAN	43.39	29.35	12.28	93.04	76.13	41.53
VLANet	45.24	31.83	14.17	95.70	82.85	33.09
MARN	48.55	31.94	14.81	90.70	70.00	37.40
CCL	-	33.21	15.68	-	73.50	41.87
RTBPN	60.04	32.36	13.24	97.48	71.85	41.18
LoGAN	51.67	34.68	14.54	92.74	74.30	39.11
VCA	58.58	38.13	19.57	98.08	78.75	37.75
LCNet	59.60	39.19	18.87	94.78	80.56	45.24
CWSTG	43.31	31.02	16.53	95.54	77.53	41.91
CPL	66.40	49.24	22.39	96.99	84.71	52.37
CRM*	53.66	34.76	16.37	-	-	-
CNM*	60.39	35.43	15.45	-	-	-
IRON*	**70.71**	**51.84**	25.01	98.96	**86.80**	**54.99**
PPS	69.06	51.49	**26.16**	**99.18**	86.23	53.01

Table 3: Performance comparisons on the Charades-STA. The best results and second best results are represented as bold and underlined numbers, respectively. The methods using additional annotations or large-scale pre-trained models are marked with *.

Ablation Study

For a more in-depth understanding of the proposed method, we perform ablation studies on our components.

Analysis on the Gaussian mixture proposal. As shown in Tab. 2, we study the impact of the different strategies to generate Gaussian mixture proposals for positive proposals. The results are summarized as follows: First, the Gaussian mixture proposals are more effective than the single Gaussian proposal, which means that the mixture proposal can better represent a query-relevant temporal location. Second, learning multiple centers and one width for one mixture proposal performs best. We conjecture that learning multiple widths makes it complicated to learn proposals, which reduces performance. Third, importance weighting from the reconstructor yields the best result by representing the importance of each mask for query reconstruction. On the other hand, importance weighting from the generator is less effective, because it is hard to reflect reconstruction-aware information.

# masks	R@1 mIoU	R@5 mIoU
Fix to 1	33.33	54.85
Fix to 3	35.91	56.27
Fix to 5	36.58	54.36
Fix to 7	36.25	49.48
Vary	**37.59**	**58.78**

(a) The number of masks for a positive proposal.

Pulling strategy	R@1 mIoU	R@5 mIoU
All	35.41	56.87
To mid	35.83	**58.92**
Distant	**37.59**	58.78

(b) Strategies for the pulling loss

Table 4: Ablation studies on the ActivityNet Captions dataset.

Fig. 3 shows the impact of the number of proposals K. The performance increases until the number is 5 at R@1,mIoU. We observe that defining too many proposals makes the proposals redundant and have short lengths due to the impact of the inter-pushing loss $\mathcal{L}_{push}^{inter}$.

Impact on a varying number of masks. For positive proposals, we form $\mathbf{P}_p^{(k)}$ by a Gaussian mixture of $E_p = k$ Gaussian masks to reflect a varying number of Gaussian masks in each positive proposal. To verify the effectiveness of the varying number of Gaussian masks, we compare the performance of fixing the number of Gaussian masks for every positive proposal in Tab. 4a. The results show that using a varying number of Gaussian masks for each positive proposal performs better than using a fixed Gaussian number of Gaussian masks. We find that combinations of different numbers of Gaussian masks can represent a diverse number of query-relevant events.

Effect of the pull-push learning scheme. In Tab. 5, we verify the effectiveness of our pull-push learning scheme. Among combinations of three losses ($\mathcal{L}_{pull}$, $\mathcal{L}_{push}^{intra}$, $\mathcal{L}_{push}^{inter}$), adopting all three losses yields the best performance. We conjecture that our pull-push learning scheme helps Gaussian masks to capture diverse events for better representing a temporal location. It is notable that adopting only the pulling loss can yield competitive or higher results to the state-of-the-art methods in Tab. 1. If the pulling loss $\mathcal{L}_{pull}$ is excluded, the performance decreases significantly. We observe that Gaussian masks for one Gaussian mixture proposal are spread sparsely throughout the entire video without $\mathcal{L}_{pull}$,

Loss			R@1,IoU=		R@5,IoU=	
$\mathcal{L}_{pull}$	$\mathcal{L}_{push}^{intra}$	$\mathcal{L}_{push}^{inter}$	0.3	mIoU	0.3	mIoU
✗	✗	✗	45.23	30.98	67.86	47.75
✓	✗	✗	55.03	36.79	72.49	51.14
✗	✓	✗	45.69	29.83	71.51	49.69
✗	✗	✓	23.50	15.96	72.30	44.46
✗	✓	✓	23.47	16.38	66.15	38.20
✓	✗	✓	41.51	30.23	85.18	**60.44**
✓	✓	✗	49.98	32.46	80.54	55.26
✓	✓	✓	**59.29**	**37.59**	**85.54**	58.78

Table 5: Ablation studies of different losses for the pull-push learning scheme on the ActivityNet Captions dataset.

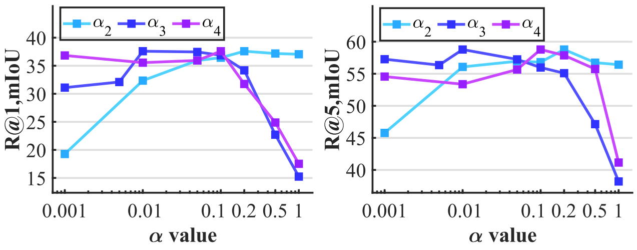


Figure 4: Ablation studies by varying α values for the pull-push learning scheme on the ActivityNet Captions dataset.

which can not represent one proper temporal location. Additionally, the results suggest that two pushing losses ($\mathcal{L}_{push}^{intra}$, $\mathcal{L}_{push}^{inter}$) are used with $\mathcal{L}_{pull}$ for a synergy effect, because the goal of the pushing losses is to make less overlapped masks for moderate coupling. For a more in-depth understanding of the pulling loss $\mathcal{L}_{pull}$, we conduct ablation studies of different strategies for $\mathcal{L}_{pull}$ in Tab. 4b. Among the strategies, pulling two distant masks closer or pulling two distant masks to the middle mask performs best. The results imply that pulling fewer masks is better and pulling more masks may ruin the structure of the mixture proposal due to overlapped masks. Fig. 4 presents the impact of controlling the balance of the losses. The results show that a high α_2 value for $\mathcal{L}_{pull}$ is needed to produce densely generated masks and the adequate α_3 and α_4 values for $\mathcal{L}_{push}^{intra}$ and $\mathcal{L}_{push}^{inter}$ are needed to cause proper discrimination between the masks and between the proposals, respectively.

Qualitative Results

Fig. 5 shows qualitative results of our PPS and other variants of PPS. It is notable that PPS captures accurate query-relevant locations, while the ground truth, which can be noisy due to the subjectivity of annotators, includes redundant locations such as a logo at the beginning of the video.

5 Conclusion

For weakly supervised temporal video grounding, we have proposed Gaussian mixture proposals with a pull-push learning scheme to capture diverse events. We express arbitrary shapes of a temporal location by learning importance, centroid, and range of every Gaussian in the mixture. To produce moderately coupled Gaussians in the mixture, we leverage a pulling loss and a pushing loss, each of which plays an opposite role to the other. Through experimental comparisons and extensive ablation studies, we have verified that our method generates multiple high-quality proposals, which greatly improve recall rates.

Limitations. We use the proposals with the shape of a Gaussian mixture, but other shapes could be explored to represent complex temporal structures.

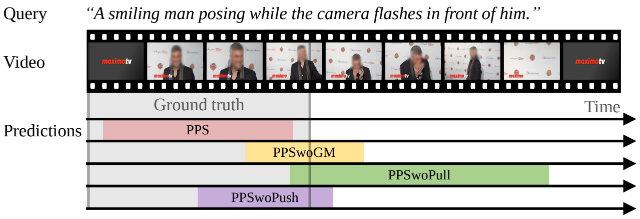


Figure 5: Qualitative results on the Activity-Net Captions dataset. Given a video and a query, PPS yields a predicted temporal location (red). We also visualize the predictions of variants using a positive proposal of one Gaussian mask without the mixture (yellow) or excluding a pulling loss (green) or excluding pushing losses (purple).

Acknowledgments

This work was supported by IITP grant funded by Korea government(MSIT) [No.B0101-15-0266, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis; NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)] and the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2023.

Placeat dicta in excepturi alias tempore sed voluptate, alias consectetur aperiam ducimus odio minima voluptas, quia reiciendis mollitia distinctio veniam repudiandae corporis.Voluptatum illo nobis provident rerum eos officia natus similique voluptates, eius repellat quia magnam itaque optio expedita tenetur molestias, libero nesciunt suscipit numquam delectus voluptatibus explicabo perferendis pariatur, voluptates eum natus possimus tempore quo deserunt, nihil eos necessitatibus magni voluptate dolores vitae vel libero.Eum distinctio nobis totam rerum praesentium officia magni incidunt fuga, nobis dolor iste nesciunt at nisi pariatur facilis vero nemo, at doloribus reprehenderit sequi consequuntur libero sint esse, consequatur explicabo nemo nobis fuga quidem libero accusantium tempore ex optio, corporis eveniet enim aspernatur quasi cupiditate?Molestias mollitia odio eligendi voluptatum aliquam, natus facilis quasi reiciendis libero sed voluptatibus, optio architecto eius dolore aut excepturi, odio quis culpa eius natus placeat consectetur id voluptate deleniti quidem, atque facilis nesciunt expedita obcaecati soluta quibusdam itaque quis magnam minima?Tempora placeat alias excepturi ipsa unde neque vel quam fugit, aut quis modi fugiat necessitatibus quam amet officia ipsum unde reiciendis?Architecto maiores suscipit debitis dolore non animi, odit enim earum explicabo tempore officiis.Soluta quae consectetur autem, labore sunt tenetur quod at quisquam officiis, neque ea quas dolores?