Efficient Video Grounding with Which-Where Reading Comprehension

Jialin Gao, Xin Sun, Bernard Ghanem, Xi Zhou, Shiming Ge, Senior Member, IEEE

Abstract—Video grounding aims at localizing the temporal moment related to the given language description, which is very helpful to many cross-modal content understanding applications like visual question answering and sentence-video search. Existing approaches usually directly regress the temporal boundaries of an event described by a query sentence in the video sequence. This direct regression manner often encounters a large decision space due to diverse target events and variable video durations, leading to inaccurate localization as well as inefficient grounding. This paper presents an efficient framework termed from which to where to facilitate video grounding. The core idea is imitating the reading comprehension process to gradually narrow the decision space, in what we decompose the direct regression into two steps. The “which” step first roughly selects a candidate area by evaluating which video segment in the predefined set is closest to the ground truth. To this end, we formulate this step into a multi-choice reading comprehension problem and propose a criterion to select the best-matched segment. In this way, the excessive decision space is effectively reduced. The “where” step aims to precisely regress the temporal boundary of the selected video segment from the shrunk decision space. We thus introduce a triple-span representation for each candidate video segment to use the regional context for better boundary regression. The “which” and “where” steps can be combined into a unified framework and learned end-to-end, leading to an efficient video grounding system. Extensive experiments on Charades-STA, ActivityNet-Captions, and TACoS benchmarks clearly demonstrate the effectiveness of our framework.

I. INTRODUCTION

UNDERSTANDING video content with the help of natural languages, such as describing the events in videos through language sentences or visual language, is essential to move towards general artificial intelligence. As a typical interdisciplinary task, video grounding [1–4] has gradually attracted the attention of academia and industry, due to its various applications in video question answering [5–7], video storytelling [8], video retrieval [9–11] and so on. The goal of video grounding is to locate a temporal span from an untrimmed video according to a given language query (shown in Fig. 1 (a)). It requires a simultaneous understanding of linguistic meaning, visual clues, and vision-language correspondence. The vision and language communities have also attempted different technical routes to solve this problem.

The computer vision community usually follows the direct regression paradigm to design the video grounding system with the aid of hand-crafted anchors [1, 2, 12] or reference points [13–15]. Sharing the same spirit of anchor-based [16] object detectors, one kind of technical route directly regresses the temporal boundaries of fixed-size bounding boxes. Similar to the anchor-free object detection methods [17], another technical way predicts the distance from the reference point to the ground-truth boundaries. To achieve accurate video grounding,
they seek to learn better vision-language representations [18] to facilitate the above direct regression process. However, the performance of these routes is strictly constrained by multiple factors, e.g., the number of anchors/reference points and the quality of vision-language representation. In particular, when some frames lack sufficient distinguishing information, their regression or prediction would suffer from adverse effects.

Actually, paying much more attention to cross-modal fusion in their solutions does not provide explicit clues for the exact target span’s location. The direct regression paradigm still faces the challenge of determining two specific values (i.e., start and end timestamps) from a large-scale decision space. It makes the regression process difficult, leading to inaccurate grounding results. On the contrary, the natural language processing community adopts the proposing-and-ranking paradigm. It mimics machine reading comprehension solutions to discretize the above continuous decision space into afinite search set. For example [19–22], they first propose a large number of video segments and then select one candidate segment according to their ranking scores. However, this paradigm brings in two notorious drawbacks. First, densely sampled proposals are required to achieve a high recall, significantly increasing the computation cost and localizing time in the grounding system. Second, these methods neglect the precision of segment boundaries because they cannot guarantee the complete coverage of all segments in the video.

To tackle the challenges mentioned above, we manage to address video grounding in a fashion from which to where. This proposed paradigm decomposes the direct regression into two sub-processes that are readily optimized, effectively narrowing the decision space and improving grounding accuracy. It consists of two steps: coarse segment selection (which) and fine-grained boundary adjustment (where). The first step selects a video segment from the candidate set, and the second step predicts offsets to refine its boundaries. Our solution benefits from this design, inheriting the advantages of the two communities’ methods while avoiding their shortcomings.

In the “which” step, we reformulate video grounding into a multi-choice video reading comprehension problem by introducing an answer generator to sample video segments. The core challenge of this step becomes how to correctly select a video segment as the candidate area. We propose to measure the degree of cross-modal semantic matching and the quality of boundary localization as selection criteria. Although this step shows a bit similar form to the proposing part in previous solutions, it inherently contains distinct characteristics. In particular, our video segments are not fixed-size but prediction-aware. The closer a video segment is to the ground truth, the less punishment is given to the corresponding prediction. Furthermore, a suppression branch is used to guarantee the frame-level distinguishing feature for segment selection by contrasting the query-related content with backgrounds.

The “where” step introduces a triple-span representation for the candidate video segment: start, center, and end regions. We aggregate the context of center region to measure the degree of semantic matching while adopting the rich information in the start and end regions to predict the boundary offsets. Due to the first step having narrowed the decision space, this step is facilitated to predict more precise offset values to refine the coarsely located area. So far, we have obtained a which-where reading comprehension framework (shown in Fig.2) by integrating these two steps into a unified end-to-end approach.

To sum up, our main contributions are three-fold. (1) We formulate video grounding as a multi-choice video reading comprehension problem and propose a novel end-to-end framework to address it. (2) We propose a suppression branch and a triple-span representation to facilitate the segment selection and boundary refinement. (3) We conduct a comprehensive evaluation on three popular benchmarks to show the effectiveness of our approach.

II. RELATED WORK

Video Grounding. This task was introduced by [1, 2] to locate relevant moments given a language query, similar to temporal action localization [23–25]. Existing solutions can be grouped into two categories: proposal-free [18, 26–28] and proposal-based [19–21]. Yuan et al. [18] builds a proposal-free method with BiLSTM and leverages the attention mechanism [12, 18, 29] to regress the temporal coordinates. Ghosh et al. [28] and Chen et al. [30] seek to cross-model interactions for two boundary regressions. Inspired by anchor-free object detection [17], Lu et al. [14] and Zeng et al. [15] propose a dense bottom-up framework to regress the distances to the start and end boundaries for each timestamp in the target video span and selects the one with the highest confidence as the final result. Mun et al. [31] effectively predicts the target time interval by exploiting contextual information from local to global during cross-modal interactions. Li et al. [32] proposes a pyramid network to explore a hierarchical architecture based on multi-scale 2D correlation maps with different temporal scales. Liu et al. [33] incorporates both local and global contexts into features of each start/end position for biaffine-based localization. They usually suffer from inferior performance due to facing a large-scale decision space in regression and ignoring the segmental context [34–36].

From the perspective of proposal usage, reinforcement learning (RL) based methods can also be viewed as proposal-free methods. RL-based methods tend to formulate Video Grounding task as a problem of sequential decision and utilize reinforcement learning to tackle it. Specifically, He et al. [37] controls an agent to read the query, to watch the video, and then to move the temporal boundaries progressively for the best-matched moment. STRONG [38] employs a dual-level reinforcement learning which considers both spatial-level and temporal-level localization. AVMR [39] formulates the task as an adversarial learning problem with two tightly connected components, i.e., a reinforcement learning as a generator and a pairwise ranking model as a discriminator. MABAN [40] employs a multi-agent reinforcement learning framework to decompose video grounding into iteratively localizing the two temporal boundary points for each query sentence. However, due to the limitation of predefined action space and the difficulty of optimizing, the performance of RL-based methods is unsatisfactory and unstable.

Most proposal-based methods transfer the continuous regression problem into the discrete classification problem via a
proposing-and-ranking pipeline. They usually generate a large number of video moment candidates by a proposal network and rank them. Gao et al. [2], and Hendricks et al. [1] adopt the sliding windows as proposals and then calculate the similarity between each proposal and the query sentence in a joint embedding space. Some solutions focus on modelling interaction between visual and textual modalities, e.g., query-aware video representation [41], query-guided proposals [13], visual-aware language embedding [42] and co-attention [30, 43].

To improve the accuracy of ranking, some prior works [19–21] explicitly model relations among proposals by 2D-map or sliding windows as proposals and then calculate the similarity between each proposal and the query sentence in a joint embedding space. Some solutions focus on modelling interaction between visual and textual modalities, e.g., query-aware video representation [41], query-guided proposals [13], visual-aware language embedding [42] and co-attention [30, 43].

To improve the accuracy of ranking, some prior works [19–21] explicitly model relations among proposals by 2D-map or graphs. Liu et al. [44] considers both cross- and self-modal relations in a joint framework to capture much higher-level interactions. Wang et al. [45] leverages the inherent structure of the moment constructed with boundary and content for both vision-language understanding and video comprehension. However, they suffer from two notorious drawbacks: heavy computation cost (i.e., calculating all moment-sentence pairs) and heuristic rules sensitivity (i.e., the temporal scales or the scales of sliding-window). To guarantee a high recall, they are required to design the predefined video moments carefully and densely place candidate proposals.

Unlike these two types of methods, we solve the problem in a fashion from which to where: first selecting the best-matched segment from a predefined set and then refining its boundaries.

**Machine Reading Comprehension.** Given the reference document or passage, MRC requires the machine to answer questions about it [46]. There are two types of the existing MRC variations related to video grounding, i.e., span extraction and multi-choice. The former [47] extracts spans from the given passage texts and has been explored by some previous works [14, 22, 28]. The latter [48, 49] aims to find only one choice in the given candidate set based on the given passage.

Motivated by them, our scheme first coarsely selects one candidate segment by formulating video grounding as cross-modal multi-choice reading comprehension. This step aims to answer the question of “which.” After that, our scheme predicts the boundary offset for the selected segment to answer the question of “where.”

### III. METHODOLOGY

#### A. Problem Definition

Multi-choice reading comprehension aims to find the correct choice from a given set based on the passage content. Referring to this definition, we treat the video \( V \) as the visual passages, the query \( Q \) as the question description, and provide a set of video segments as answer choices \( A \). Video grounding is similar to Multi-choice Reading Comprehension (MRC), based on the given triplet \((V, Q, A)\).

However, the essence of video grounding is to answer where is the location of a query-related event in video, rather than which choice is the best like MRC. It is not rigorous to directly reformulate video grounding into multi-choice reading comprehension. Thus, we modify this formulation as a consistent video reading comprehension for addressing video grounding in a fashion from which to where. Specifically, we first select the best-matched segment choice corresponding to the given query among all candidates and then refine the boundary of the selected one for accurate localization.

We have one natural language sentence and an associated ground-truth for each query-video pair, where \((g^s, g^e)\) denote the start and end timestamps of the target video span. Each language query sentence is represented by \( Q = \{q_i\}_{i=1}^L \), where \( L \) is the number of tokens. The untrimmed video is denoted as \( V = \{v_1, v_2, \cdots, v_n\} \in \mathbb{R}^{n \times C} \), where \( n \) is the number of frames. We employ an answer generator to provide \( N^* \) video segments as answer choices \( A = \{a_1, \cdots, a_{N^*}\} \), where each segment \( a_i = (\tau_i^s, \tau_i^e) \) contains the start \( \tau_i^s \) and end \( \tau_i^e \) timestamps. After these notations, the reformulation
is composed of two steps: coarse segment selection and fine-grained boundary refinement:

\[
\hat{a} = \arg\max_{a_i \in A} P(a_i | (V, Q, A)), \tag{1}
\]

\[
\hat{a}^* = \text{Boundary}(\hat{a}),
\]

where \(\text{Boundary}(\cdot)\) denotes the boundary refinement, \(\hat{a}\) and \(\hat{a}^*\) are the selected and refined video segment, respectively.

### B. Architecture

Our framework mainly consists of three components: vision-query alignment, answer generator, and grounding module, as illustrated in Fig. 2. Specifically, we describe the details of each component in our framework as follows:

1) **Vision-query Alignment**: This module aims to encode the content of language query sentences and video by two separate encoders. Each encoder is specifically designed to aggregate the intra-modality context from each frame or token feature. Details are introduced as follows:

- **Vision Encoder**: We first adopt a temporal 1D convolution to map the CNN-extracted feature sequence \(V\) to the desired dimension, which is followed by an average pooling layer to reshape the sequence into the desired length \(T\). Then, we employ three stacked 1D convolution layers with ReLU units, denoted as Conv1D, to model local temporal relations. Hence, we have encoded visual features as \(V \in \mathbb{R}^{T \times C}\).

- **Query Encoder**: Each word \(q_i\) of query \(Q\) is initially represented with the embedding vectors by the GloVe-300B [50]. Then we sequentially feed the initialized embedding into a three-layer Bi-LSTM network to capture semantic information and temporal context. We take the last layer’s hidden state as \(\hat{Q} \in \mathbb{R}^{L \times C}\) for cross-modality alignment with video representation \(\hat{V}\).

These two independent encoders can be denoted as follows:

\[
\hat{V} = \text{VisionEncoder}(V), \tag{2}
\]

\[
\hat{Q} = \text{QueryEncoder}(Q).
\]

2) **Alignment**: We separately use a simple 1D temporal convolution layer to project visual features and query representations to a common space. A MaxPooling layer squeezes the query feature to a single vector with dimension \(1 \times C\). Similar to recent methods [15, 19, 31], the Hadamard product is used to align the visual and linguistic features, followed by \(L_2\) normalization. Although there are more advanced fusion or alignment methods, we chose this simple way to evaluate the generality of our model. The output of this module is denoted as \(F = \{f_i\}_{i=1}^{T} \in \mathbb{R}^{T \times C}\).

3) **Answer Generator**: The answer generator in our model aims to provide a set of video segments. Similar to related works [19, 21], we also denote a video segment in the answer set \(A\) as a pair of its starting and ending timestamps so that \(a_i = (\tau_i^s, \tau_i^e)\). However, we construct segment-level features based on the regional characteristics of each candidate choice by a temporal RoI align, which is the 1D implementation of RoI align used in object detection [51].

Without loss of generality, we first divide the video \((V)\) into \(N\) clips, and then we have a temporal proposal map with a temporal scale \((1/N)\). We show an example of \(N = 8\) in Fig. 3. The horizontal and vertical axes indicate the starting and ending clips indices. The video segment, denoted as darkest one, ranges from the \(2^{nd}\) clips to \(6^{th}\) clips, means that the normalized temporal range is \((2/8, 6/8)\). White blocks indicate all the invalid choices in the left-bottom, where the starting boundaries exceed the ending boundaries. Thus these choices (white blocks) are invalid and not considered during training and inference. All of the valid choices compose the answer set \(A\). For a valid choice, it is represented as three regional features, which are sampled from the modulated video sequence \(\hat{F} \in \mathbb{R}^{T \times C}\) according to the coordinates. \(N_s, N_c\) and \(N_e\) indicate the length of sampling feature.

![Fig. 3. Illustration of segment choices (left) and triple representation construction of choice \(a_i\) (right), the best-matched one denoted with the darkest color. Each segment choice is illustrated by a square block. Choices in the same row/column share the same start/end time. The starting boundaries of choices at the left-bottom exceed the ending boundaries. Thus these choices (white blocks) are invalid and not considered during training and inference. All of the valid choices compose the answer set \(A\). For a valid choice, it is represented as three regional features, which are sampled from the modulated video sequence \(\hat{F} \in \mathbb{R}^{T \times C}\) according to the coordinates. \(N_s, N_c\) and \(N_e\) indicate the length of sampling feature.](image)

![Fig. 4. The detailed illustration of our suppression branch in Fig. 2. The input feature sequence is modulated by the foreground weights generated from three stacked temporal convolution layers. Each layer is denoted as \(\text{Conv}(c_f, c_k, \text{Act})\), where \(c_f, c_k\) and \(\text{Act}\) are filter numbers, kernel size and activation function. \(\otimes\) depicts the multiplication modulation.](image)
where $F$ is the output of vision-query alignment, $\hat{F}$ is the modulated feature sequence, and $\otimes$ denotes the element-wise multiplication with broadcasting.

- **Triple-span representations.** We illustrate the schematic of triple-span representation in the right part of Fig. 3. Specifically, we first expand the boundary regions of each segment with an expansion ratio $\eta$ for the representation construction. Motivated by [22, 52], we then employ the RoI align operation for sampling fixed-length features. Given a video segment $(\tau_s^j, \tau_e^j)$, we denote the starting and ending regions as $(\tau_s^j - \eta d, \tau_e^j + \eta d)$ and $(\tau_s^j - \eta d, \tau_e^j + \eta d)$, respectively, where $d = (\tau_e^j - \tau_s^j)$. The center region is defined as $(\tau_s^j, \tau_e^j)$. Based on these division, a temporal RoI Align operation is used to construct the feature representation of the video segment choice. So far, we have the $F_A$ for the answer set $A$. We use $N$ to indicate the total length of a choice representation, where $N = N_s + N_e + N_c$. The first two parts are used to predict the boundary offsets for refinement. The last one is to evaluate the matching quality between the segment choice and the query.

- **Localization branch.** This section illustrates how to achieve video grounding from *which* to *where*. We first select the best-matched choice from a candidate set and then refine it’s boundaries by boundary regression. Therefore, this branch is responsible for both segment selection and boundary refinement. Taking the choice representations as input, this module predicts the semantic matching score $p_i \in P_A$ and boundary offsets $(\delta_s^j, \delta_e^j)$ for the choice $\alpha_j$. Based on these predictions, we define the best-matched choice according to the degree of semantic matching and the quality of boundary localization.

Specifically, with the modulated feature sequence $\hat{F}$ from the suppression branch, we first adopt three stacked temporal convolution layers with ReLU units to enhance it by modeling the local temporal relation. Then, the enhanced feature is combined with the original feature $F$ in a residual-learning form by a skip-connection. Afterward, the feature sequence is used to generate regional features $(F_s, F_c, F_e)$ for the triple-span representations $F_A$ by a temporal RoI Align operation [51]. Afterward, these regional features are fed into the matching head and regression head. These two heads share the same architecture, where they contain two stacked temporal convolution layers with a ReLU unit and one sigmoid activation layer. Detailed architectures are illustrated in Fig. 5 for better understanding.

For regional feature $F_c$, our matching head predicts a score $p^m$ (index $i$ is omitted for better readability) to measure the degree of semantic matching between the content of video segment choice and the linguistic meaning of the query. For regional features $F_s$ and $F_e$, our regression head predicts the starting and ending boundary offsets, respectively. Hence, we have the coarse segment selection criterion as:

$$L_{select,i} = L_{sem,i} + L_{loc,i},$$  \hspace{1cm} (4)

where $L_{select,i}$ is the cost of selecting $i$-th segment as the best-matched one for the ground truth $(g_s^i, g_e^i)$. In Eq. 4, $L_{sem,i}$ and $L_{loc,i}$ are used to measure the degree of semantic matching and the quality of boundary localization, respectively. These two items can be written as:

$$L_{sem,i} = L_{bce}(p_{i}^{m}, \mathbb{1}),$$
$$L_{loc,i} = L_{1}(\tau_s^i, g_s^i) + L_{1}(\tau_e^i, g_e^i),$$  \hspace{1cm} (5)

where $L_{bce}$ indicates the binary cross-entropy loss function for calculating the degree of matching and $L_{1}$ loss is adopted to measure the localization quality between $i$-th choice $(\tau_s^i, \tau_e^i)$ and the ground truth $(g_s^i, g_e^i)$. The segment with the smallest value among $L_{select}$ is selected as the best-matched choice. We denote it as $j$-th segment choice in $A$ for simplification. Then, the boundary refinement of it can be computed as:

$$t_s^j = \tau_s^i + \delta_s^j,$$
$$t_e^j = \tau_e^i + \delta_e^j,$$  \hspace{1cm} (6)

where $\delta_s^j$ and $\delta_e^j$ are the predicted starting and ending offsets of the best-matched one $(\tau_s^j, \tau_e^j), t_s^j$ and $t_e^j$ are the final grounding results.
Notably, the coarse segment selection and fine-grained boundary adjustment in our grounding module are combined into a unified training approach. Unlike prior works, our video segments in the answer set are prediction-aware. That is because we predict the boundary offsets for each segment choice in each iteration based on the parameters optimized in the previous iteration. In turn, the selected video segment is defined as the positive training sample for supervising the regression in the fine-grained boundary adjustment process.

C. Training and Inference

In this section, we are going to introduce the training and inference stages for our solution in detail. In the beginning, we describe the overview of our objective function used in model optimization. Then, we make a detailed explanation for each component and define the corresponding label assignment for the training stage. Afterward, we illustrate how to generate grounding results in the inference stage.

1) Training stage: In general, our objective function consists of three losses, which can be formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{supp}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{sem}},$$

where $\mathcal{L}_{\text{supp}}$ is the item from the suppression branch, $\mathcal{L}_{\text{reg}}$ and $\mathcal{L}_{\text{sem}}$ are from the localization branch.

Comparing with previous methods [18–20], we introduce auxiliary supervision ($\mathcal{L}_{\text{supp}}$) for modulating the feature response in the suppression branch. The regression ($\mathcal{L}_{\text{reg}}$) and semantic matching ($\mathcal{L}_{\text{sem}}$) losses are used to evaluate the quality of localization and the degree of the choice-query matching, respectively. In our scheme, we dynamically define the positive sample for optimization through this measurement item $\mathcal{L}_{\text{select}}$ rather than the predefined/manual way they used.

For the label assignment in the suppression branch, we empirically refer to the way in BSN [52] and expand the length of ground truth instance by a ratio $\phi$. For a ground truth instance $(g^s, g^e)$, the label of each temporal location lying in the expanded region $(g^s - d/\phi, g^e + d/\phi)$, where $d = g^e - g^s$, will be set to 1, otherwise 0. Hence, the item from the suppression branch can be written as:

$$\mathcal{L}_{\text{supp}} = \frac{1}{T} \sum_{i=1}^{T} \left( -\log(p_i^T) + (1 - \delta_i) \log(1 - p_i^T) \right),$$

where $T$ is the length of the generated foreground probability sequence $F_{fg}$ and $\Delta_i$ indicates the binary label assigned in the above label assignment.

For the label assignment in the localization branch, we first determine the best-matched segment choice (i.e., $j$-th choice in answer set $A$) as the positive sample according to the measurement item $\mathcal{L}_{\text{select}}$. We then use the $L_1$ loss to supervise the predicted boundaries $(t_j^s, t_j^e)$ of the selected one with the ground truth $(g_j^s, g_j^e)$ as:

$$\mathcal{L}_{\text{reg}} = L_1(t_j^s, g_j^s) + L_1(t_j^e, g_j^e),$$

where $\delta_j^s$ and $\delta_j^e$ are the predicted start and end offsets of the best-matched choice ($\tau_j^s, \tau_j^e$) in Fig. 2.

Since only one choice is selected as a positive sample for each ground truth instance, our formulation in Eq. 1 will bring the issue of imbalanced training samples on semantic matching evaluation, which usually causes difficulty in optimization and further limits the performance of our model. To this end, we resort to the Focal loss ($\mathcal{F}_L$) in [53] to alleviate this issue. And the hyperparameters $\alpha$ and $\gamma$ of Focal loss are 0.25 and 3.0, respectively. The item $\mathcal{L}_{\text{sem}}$ is defined as:

$$\mathcal{L}_{\text{sem}} = \frac{1}{N^*} \sum_{i=1}^{N^*} \mathcal{F}_L(p_i^m, y_i),$$

where $N^* = N(N-1)/2$ denotes the number of choices, and $N$ is the number of clips defined in III-B2, $y_i$ is the label of the $i$-th candidate choice, 1 for positive and 0 for negative.

Feature Extractor. Since different feature extractors would cause dramatic effects on the grounding performance, we use the most popular feature extractor [54–57] in previous methods for fair comparison. Following [20, 58], we adopt VGG [59] feature for the Charades-STA and C3D [56] for the TACoS and ActivityNet-Captions datasets. Considering some state-of-the-art approaches [15, 22, 28] using a better extractor, we also utilize the I3D [57] feature to make comparisons on the Charades-STA benchmark. For language feature extractor, we use the 300d GloVe [50] word embedding model as most previous solutions do, like [30, 60].

Architecture settings. In all experiments, we set the hidden units of all Bi-LSTM layers to be 256, and $T$ in the visual encoder is defined as 128 for the TACoS dataset, 64 for the ActivityNet-Captions dataset, and 16 for the Charades-STA dataset. The dimension $C$ of channels is 512. We adopt position embedding in the ActivityNet-Captions benchmark as [15, 21], which is used just after the average pooling layer in the visual encoder. For the choice generator, we set the number of clips $N$ in section III-B2 as 25 for the TACoS dataset, 20 for the ActivityNet-Captions dataset, and 12 for the Charades-STA dataset. For the sampling length in temporal RoI pooling operation, we set the total number $N_t$ to be 32, where $N_c$, $N_r$ and $N_t$ are 8, 16, and 8, respectively. The extension ratio $\eta$ is set as 0.2. We also conduct experiments in the next section about exploring the impact of the extension ratio $\eta$ on the final grounding performance.

2) Inference stage: During the inference, we cannot use the location information of ground-truth instances to select the best-matched choice, so that we set the value of $\mathcal{L}_{\text{loc}}$ in Eq. 4 to be 0. We choose the one with the highest matching score as the final prediction to perform video grounding.

Considering previous solutions compare their performance of top-1 and top-5 predictions, we also evaluate our proposed approach in this condition. Our approach first selects 5 best-matched choices and calculates the starting and ending offsets for every choice. The selected 5 candidate segments are denoted as $\{p_i^m, t_i^s, t_i^e\}^5_{i=1}$, where $p_i^m, t_i^s, t_i^e$ represent the visual-language matching score, the start, end time of the refined candidate choice $a_i$, respectively.
IV. EXPERIMENTS

A. Experimental Setting

Datasets. We conduct experiments on three popular benchmarks: TACoS [61], Charades-STA [63], and ActivityNet-Captions [62], to verify the effectiveness of our approach. The TACoS consists of 127 videos on cooking activities and has 18818 video-query pairs in total. Following the standard split [2], we adopt 10146, 4589, and 4083 video-query pairs for training, validation, and testing, respectively. The Charades-STA is originally designed for action recognition and then extended by Gao et al. [2] with clip-level sentence annotations in a semi-automatic way. It contains 6672 videos as well as 12,408 and 3,720 query-video pairs for the training and testing set, respectively. The ActivityNet-Captions consists of 19,209 videos in total, where the training and testing set has 10,099 and 4917 videos, respectively. Following the setting in previous methods including CMIN [58], DRN [15] and 2D-TAN [19], we also use val_1 as validation set and val_2 as testing set, which have 37,421, 17,505, and 17,031 query-video pairs for training, validation, and testing, respectively. Detailed statistics of these benchmarks are shown in Tab. I.

Evaluation metric. For a fair comparison, we follow the way in Gao et al. [2] to compute the Rank $k@\mu$ as the convention. It denotes the percentage of testing samples that have at least one correct answer in the top-k choices. The refined choice $\hat{\eta}_i$ is correct when its temporal Interaction-over-Union (IoU) $\hat{\delta}_i$ with the ground-truth is larger than a threshold $\mu$; otherwise, the choice is wrong. Specifically, we set $k \in \{1, 5\}$ and $\mu \in \{0.1, 0.3, 0.5\}$ for the TACoS dataset and $\mu \in \{0.5, 0.7\}$ for the Charades-STA and ActivityNet-Captions benchmarks. In addition, we also illustrate the performance difference on mIoU when analyzing the effect of extension ratio $\eta$ on TACoS and Charades-STA benchmarks.

Training settings. We adopt the AdamW optimizer and step strategy for all experiments. The weight decay is set to 1e-5. The learning rate, initialized as $6 \times 10^{-4}$, will be decayed by a factor of 10 every 4 epochs. The batch size is 32 for 2 RTX 1080TI GPU. The total training epochs is 12.

B. State-of-the-art Comparisons

Comparisons on TACoS. We report the grounding accuracy of our solution compared to previous methods, which is illustrated in Tab. II, where the outstanding performance achieved by our solution is highlighted. In general, our approach beats recent competing approaches by a significant margin across all evaluation metrics. For instance, we improve the result from 25.32% in 2D-TAN [19] to 33.78% regarding Rank $1@0.5$. Although we have employed a similar strategy to 2D-TAN [19] for generating choices, our approach can refine the boundaries of the best-matched choice, which helps us get rid of relying on densely placed choices. That is why we can obtain better grounding accuracy than 2D-TAN [19] by using only one-tenth of the size of the 2D-TAN candidate answer set. This efficiency is also illustrated in Tab. III. It is worth noting that multi-query and long video duration make this benchmark very challenging. However, compared with 2D-TAN [19], VSLNet [22], CSMGAN [44], and BPNet [69], we still achieve the highest scores and significant improvements across almost evaluation metrics. This observation indicates the effectiveness of our solution. Further, we focus on refining the boundaries of the best-matched choice, where we only

TABLE I
THE STATISTICS OF TEMPORAL LANGUAGE GROUNDING DATASETS. WE REPORT RELEVANT INFORMATION FOR EACH DATASETS AVAILABLE FOR THE GROUNDING TASK. NUM., VOCAB. AND AVE. REPRESENT THE ABBREVIATION OF NUMBER, VOCABULARY AND AVERAGE, RESPECTIVELY.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>Year</th>
<th>Num. Videos</th>
<th>Year</th>
<th>Num. Videos</th>
<th>Year</th>
<th>Num. Videos</th>
</tr>
</thead>
</table>

TABLE II
PERFORMANCE COMPARISON ON TACoS. ALL RESULTS ARE REPORTED IN PERCENTAGE (%) W.R.T RANK $1, 5@\{0.1, 0.3, 0.5\}$

<table>
<thead>
<tr>
<th>Methods</th>
<th>Rank1@0.1</th>
<th>Rank1@0.3</th>
<th>Rank1@0.5</th>
<th>Rank5@0.1</th>
<th>Rank5@0.3</th>
<th>Rank5@0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCN [1]</td>
<td>14.42</td>
<td>-</td>
<td>5.38</td>
<td>37.35</td>
<td>-</td>
<td>10.33</td>
</tr>
<tr>
<td>TGN [30]</td>
<td>41.87</td>
<td>21.77</td>
<td>18.9</td>
<td>53.40</td>
<td>39.06</td>
<td>31.02</td>
</tr>
<tr>
<td>MCF [64]</td>
<td>25.84</td>
<td>18.64</td>
<td>12.53</td>
<td>52.96</td>
<td>37.13</td>
<td>24.73</td>
</tr>
<tr>
<td>ACRN [41]</td>
<td>24.22</td>
<td>19.52</td>
<td>14.62</td>
<td>47.42</td>
<td>34.97</td>
<td>24.88</td>
</tr>
<tr>
<td>ACL-K [60]</td>
<td>31.64</td>
<td>21.47</td>
<td>18.01</td>
<td>57.85</td>
<td>42.15</td>
<td>30.66</td>
</tr>
<tr>
<td>CMIN [65]</td>
<td>32.48</td>
<td>24.64</td>
<td>18.05</td>
<td>62.13</td>
<td>38.46</td>
<td>27.02</td>
</tr>
<tr>
<td>SM-RI [66]</td>
<td>26.51</td>
<td>20.25</td>
<td>15.95</td>
<td>50.01</td>
<td>38.47</td>
<td>27.84</td>
</tr>
<tr>
<td>DEBUG [14]</td>
<td>-</td>
<td>23.45</td>
<td>11.72</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SLTA [67]</td>
<td>23.13</td>
<td>17.07</td>
<td>11.92</td>
<td>46.52</td>
<td>32.90</td>
<td>20.86</td>
</tr>
<tr>
<td>ABLR [18]</td>
<td>34.70</td>
<td>19.50</td>
<td>9.40</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2D-TAN [19]</td>
<td>47.59</td>
<td>37.29</td>
<td>25.32</td>
<td>70.31</td>
<td>57.81</td>
<td>45.04</td>
</tr>
<tr>
<td>VSLNet [22]</td>
<td>-</td>
<td>29.61</td>
<td>24.27</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SM-Rl [66]</td>
<td>34.51</td>
<td>20.25</td>
<td>15.95</td>
<td>50.01</td>
<td>38.47</td>
<td>27.84</td>
</tr>
<tr>
<td>ABLR [18]</td>
<td>34.70</td>
<td>19.50</td>
<td>9.40</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MCF [64]</td>
<td>25.84</td>
<td>18.64</td>
<td>12.53</td>
<td>52.96</td>
<td>37.13</td>
<td>24.73</td>
</tr>
<tr>
<td>CBLN [69]</td>
<td>49.16</td>
<td>38.98</td>
<td>27.65</td>
<td>73.12</td>
<td>59.96</td>
<td>46.24</td>
</tr>
<tr>
<td>Ours</td>
<td>51.17</td>
<td>42.51</td>
<td>33.78</td>
<td>81.67</td>
<td>68.58</td>
<td>54.22</td>
</tr>
</tbody>
</table>

TABLE III
INFERENCE TIME COMPARISON ON THE TACoS BENCHMARK.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>4909ms</td>
<td>2110ms</td>
<td>220ms</td>
<td>140ms</td>
<td>25ms</td>
<td></td>
</tr>
</tbody>
</table>

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2022.3174136, IEEE Transactions on Circuits and Systems for Video Technology

Tab. IV summarizes the Comparisons on Charades-STA. Ours also outperforms a typical proposal-free method like SMIN [45], CSMGAN [44], CBLN [69] in terms of Rank 1, especially in terms of Rank 5.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Rank1@0.5</th>
<th>Rank1@0.7</th>
<th>Rank5@0.5</th>
<th>Rank5@0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG</td>
<td>23.63</td>
<td>8.89</td>
<td>58.92</td>
<td>29.52</td>
</tr>
<tr>
<td>CTRL [2]</td>
<td>24.36</td>
<td>9.01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>AC1-K [60]</td>
<td>30.48</td>
<td>12.20</td>
<td>64.84</td>
<td>35.13</td>
</tr>
<tr>
<td>QSPN [13]</td>
<td>35.60</td>
<td>15.80</td>
<td>79.40</td>
<td>45.40</td>
</tr>
<tr>
<td>DEBUG [14]</td>
<td>37.39</td>
<td>17.69</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MCN [1]</td>
<td>17.46</td>
<td>8.01</td>
<td>48.22</td>
<td>26.73</td>
</tr>
<tr>
<td>ABLR [18]</td>
<td>24.36</td>
<td>9.01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SAP [68]</td>
<td>27.42</td>
<td>13.36</td>
<td>66.37</td>
<td>38.15</td>
</tr>
<tr>
<td>MAN [20]</td>
<td>21.42</td>
<td>20.54</td>
<td>83.21</td>
<td>51.85</td>
</tr>
<tr>
<td>2D-TAN [19]</td>
<td>39.70</td>
<td>23.31</td>
<td>80.32</td>
<td>51.26</td>
</tr>
<tr>
<td>DRN [15]</td>
<td>42.90</td>
<td>23.68</td>
<td>87.80</td>
<td>54.87</td>
</tr>
<tr>
<td>BPNet [69]</td>
<td>38.25</td>
<td>20.51</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ours</td>
<td>45.64</td>
<td>26.13</td>
<td>86.01</td>
<td>57.06</td>
</tr>
</tbody>
</table>

1D

<table>
<thead>
<tr>
<th>Methods</th>
<th>Rank1@0.5</th>
<th>Rank1@0.7</th>
<th>Rank5@0.5</th>
<th>Rank5@0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExCL [28]</td>
<td>44.10</td>
<td>22.40</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VSLNet [22]</td>
<td>54.19</td>
<td>35.22</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DRN [15]</td>
<td>53.09</td>
<td>31.50</td>
<td>89.06</td>
<td>60.05</td>
</tr>
<tr>
<td>BPNet [69]</td>
<td>50.75</td>
<td>31.64</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CPNet [32]</td>
<td>60.27</td>
<td>38.74</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LGVTI [31]</td>
<td>59.46</td>
<td>35.48</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MABAN [40]</td>
<td>56.29</td>
<td>32.26</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SMIN [45]</td>
<td>64.06</td>
<td>40.75</td>
<td>89.49</td>
<td>68.09</td>
</tr>
<tr>
<td>AVMR [39]</td>
<td>54.59</td>
<td>-</td>
<td>72.78</td>
<td>-</td>
</tr>
<tr>
<td>CSMGAN [44]</td>
<td>60.04</td>
<td>37.34</td>
<td>89.01</td>
<td>61.85</td>
</tr>
<tr>
<td>STRONG [38]</td>
<td>50.14</td>
<td>19.30</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CBLN [69]</td>
<td>61.13</td>
<td>38.22</td>
<td>90.33</td>
<td>61.69</td>
</tr>
<tr>
<td>Ours</td>
<td>60.56</td>
<td>37.86</td>
<td>90.37</td>
<td>65.32</td>
</tr>
</tbody>
</table>

CBLN [69], we obtain the competitive performance and achieve the highest score in terms of Rank 5@IoU=0.5.

Comparisons on ActivityNet-Captions. Tab. V illustrates the grounding results of our approach and other state-of-the-art methods, where we achieve competitive grounding performance over all previous solutions for all evaluation metrics. In terms of Rank 1 and high IoU threshold (i.e., 0.7), our model reaches 29.89% compared with 30.34% in SMIN [45]. SMIN [45] focuses on the complex vision-language interactions while we only adopt the simple counterpart via Hadamard product. Their strategy could be integrated into our framework for further improvement. For recent proposal-free methods, our approach outperforms the CPNet [32], LGVTI [31], and MABAN [40] by 7.99%, 7.04%, and 3.67% in terms of Rank 1@0.5, respectively. With the which-where reading comprehension, we achieve the highest score with respect to the Rank 5 and high IoU.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Rank1@0.5</th>
<th>Rank1@0.7</th>
<th>Rank5@0.5</th>
<th>Rank5@0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCN [1]</td>
<td>21.36</td>
<td>6.43</td>
<td>53.23</td>
<td>29.70</td>
</tr>
<tr>
<td>CTRL [2]</td>
<td>29.01</td>
<td>10.34</td>
<td>59.17</td>
<td>37.54</td>
</tr>
<tr>
<td>TGN [30]</td>
<td>27.93</td>
<td>-</td>
<td>44.20</td>
<td>-</td>
</tr>
<tr>
<td>ACRN [41]</td>
<td>31.67</td>
<td>11.25</td>
<td>60.34</td>
<td>38.57</td>
</tr>
<tr>
<td>ExCL [28]</td>
<td>42.7</td>
<td>24.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CMIN [65]</td>
<td>44.62</td>
<td>24.48</td>
<td>69.66</td>
<td>52.96</td>
</tr>
<tr>
<td>DEBUG [14]</td>
<td>39.72</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ABLR [18]</td>
<td>36.79</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PMI [70]</td>
<td>38.28</td>
<td>17.83</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DRN [15]</td>
<td>45.45</td>
<td>23.49</td>
<td>77.97</td>
<td>50.30</td>
</tr>
<tr>
<td>BPNet [69]</td>
<td>42.07</td>
<td>24.69</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SMIN [45]</td>
<td>48.46</td>
<td>30.34</td>
<td>81.16</td>
<td>62.11</td>
</tr>
<tr>
<td>CSMGAN [44]</td>
<td>49.11</td>
<td>29.15</td>
<td>77.43</td>
<td>59.63</td>
</tr>
<tr>
<td>CPNet [32]</td>
<td>40.56</td>
<td>21.63</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LGVTI [31]</td>
<td>41.51</td>
<td>23.07</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MABAN [40]</td>
<td>44.88</td>
<td>25.66</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CBLN [69]</td>
<td>48.12</td>
<td>27.60</td>
<td>79.32</td>
<td>63.41</td>
</tr>
<tr>
<td>Ours</td>
<td>48.55</td>
<td>29.89</td>
<td>80.95</td>
<td>63.47</td>
</tr>
</tbody>
</table>

C. Ablation Studies

To evaluate the effectiveness of each component in our solution by complete and in-depth ablation studies, we first explore the effect of repeating the “where” step in our pipeline to investigate if the proposed structure divided into more steps will be effective. And then we study the effect of the suppression branch by conducting experiments with or without it. Then, we analyze the role of boundary refinement from statistics. Afterward, we conduct ablative experiments to illustrate the impact of extension ratio η in defining the start, center, and end regions. Finally, we display some qualitative results for visualization.

1) Effect of several “where” steps. Inspired by the cascade regression in the object detection field, we consider dividing the boundary regression of the candidate choice closest to the ground truth. This formulation helps our model reach the highest score, especially in terms of Rank 1.

Our solution also outperforms other competitive methods, like SMIN [45], CSMGAN [44], CBLN [69] in terms of Rank 5. For example, we exceed the DRN [15] by 10.61% and 20.86% in terms of Rank 1@0.5 and Rank 5@0.5 respectively. Our approach also outperforms a typical proposal-free method CPNet [32] by 5.09% with respect to Rank 1@0.5.

Comparisons on Charades-STA. Tab. IV summarizes the performance comparison between our solution and other start-of-the-art methods. For VGG extractor, we almost achieve the best grounding performance across all the evaluation metrics except for that in terms of Rank 5@0.5. Specifically, our model outperforms DRN [15] by 2.74% and 2.45% absolute improvements with regards to Rank 1@0.5 and Rank 5@0.5 respectively. For the evaluation in Rank 5@0.5, our model achieved 86.01% to be inferior to that 87.80% in DRN [15]. However, when using 1D feature extractor, our solution achieves better performance than DRN [15] (i.e., 90.37% vs. 89.06%). This difference may be attributed to the bias between different visual feature embeddings from VGG or 1D extractor. We also outperform STRONG [38], AVMR [39], and CSMGAN [44] across all the evaluation metrics. Compared with proposal-free solutions, including CPNet [32], LGVTI [31], and MABAN [40], our proposed approach exceeds all of them in terms of Rank 1@0.5. Compared with advanced methods like SMIN [45],


2) Effect of background suppression: First of all, we conducted four experiments to evaluate the effect of our suppression branch and localization branch. The configurations of these experiments are as follows: 1) removing both the suppression branch and boundary regression head from our framework shown in Fig. 2; 2) only removing the suppression branch; 3) only removing the regression head; 4) maintaining these two components. The comparison results are illustrated in the Tab. VII, where Supp. and Bound. denote the abbreviation of suppression branch and boundary regression head. The symbol ✓ or ✗ represents the corresponding experiments with or without a certain component.

Tab. VII shows that the grounding performance has been significantly improved by integrating these two components. Comparing the results in the first three rows on the TACoS dataset, we can conclude that our suppression branch and boundary regression bring 4.60% and 8.36% gains in terms of Rank 1@0.3. The improvements in the ActivityNet-Captions dataset are 1.32% and 3.20% in terms of Rank 1@0.7. In summary, we get the following observations from Tab. VII:

- The first two experiments have degenerated into finding the best-matched segment among the predefined answer set. We observe the positive effect of the suppression branch on the segment selection. It promotes the network to learn better representations by background suppression, leading to a grounding performance improvement.
- Compared to the first experiment, we can witness the evident gain in all the evaluation metrics from the third experiment because of the boundary refinement. The significantly improved score proves the positive impact of our regression heads on boundary refinement.
- Compared with the first two experiments, the last two experiments indicate continuous performance improvement with the help of boundary regression. The last row in Tab. VII has reached the highest score among these four experiments, demonstrating the complementary between these two components.

3) Effect of boundary refinement: As shown in Fig. 8, we display the distribution of target video spans annotated on TACoS and Charades-STA test set in the left part and illustrate the distribution of initial and refined choices in the right part. We mark the initial segment choice, generated by the answer generator, with a light green dot while denoting the refined segment choice, from our regression head, with the red dot. From this figure, it is recognized that the positive effect of

---

**TABLE VI**

**ANALYSIS OF THE EFFECT OF REPEATING THE "WHERE" STEP ON THREE CHALLENGE BENCHMARKS. ALL RESULTS ARE REPORTED IN PERCENTAGE (%) IN TERMS OF RANK 1@0.5**

<table>
<thead>
<tr>
<th>Step</th>
<th>TACoS</th>
<th>ActivityNet-Captions</th>
<th>Charades-STA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33.08</td>
<td>46.61</td>
<td>44.41</td>
</tr>
<tr>
<td>2</td>
<td>33.51</td>
<td>47.40</td>
<td>45.01</td>
</tr>
<tr>
<td>3</td>
<td>33.78</td>
<td>48.12</td>
<td>45.43</td>
</tr>
<tr>
<td>4</td>
<td>33.49</td>
<td><strong>48.55</strong></td>
<td><strong>45.64</strong></td>
</tr>
<tr>
<td>5</td>
<td>32.88</td>
<td>48.04</td>
<td>44.98</td>
</tr>
</tbody>
</table>

**TABLE VII**

**ANALYSIS OF THE EFFECT OF BOUNDARY REGRESSION AND SUPPRESSION BRANCH, DENOTED AS Bound. AND Supp. ON TACoS AND ACTIVITYNET-CAPTIONS DATASETS. ALL RESULTS ARE REPORTED IN PERCENTAGE (%) IN TERMS OF RANK 1@µ ∈ {0.3, 0.5, 0.7}**

<table>
<thead>
<tr>
<th>Supp.</th>
<th>Bound.</th>
<th>TACoS (Rank 1@µ)</th>
<th>ActivityNet-Captions (Rank 1@µ)</th>
<th>Charades-STA (Rank 1@µ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>30.82 16.90 40.12</td>
<td>23.80</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✗</td>
<td>35.42 22.03 40.85</td>
<td>25.12</td>
<td></td>
</tr>
<tr>
<td>✗</td>
<td>✓</td>
<td>39.18 30.28 42.35</td>
<td>27.00</td>
<td></td>
</tr>
<tr>
<td>✗</td>
<td>✗</td>
<td><strong>41.78</strong> 33.08 46.61</td>
<td><strong>28.00</strong></td>
<td></td>
</tr>
</tbody>
</table>

• The first two experiments have degenerated into finding the best-matched segment among the predefined answer set. We observe the positive effect of the suppression branch on the segment selection. It promotes the network to learn better representations by background suppression, leading to a grounding performance improvement.
• Compared to the first experiment, we can witness the evident gain in all the evaluation metrics from the third experiment because of the boundary refinement. The significantly improved score proves the positive impact of our regression heads on boundary refinement.
• Compared with the first two experiments, the last two experiments indicate continuous performance improvement with the help of boundary regression. The last row in Tab. VII has reached the highest score among these four experiments, demonstrating the complementary between these two components.
our boundary refinement on the final grounding performance. This refinement facilitates the network to reduce the difference in distribution between the initial segment choices and target spans. Compared with the distribution of initial choices, our refined choices are more in line with the target distribution, which is exactly the key to performance improvement.

In order to further analyze the effect of boundary regression, we also calculate the boundary error between our predicted results and ground truth instances. The boundary error of initial and refined segment choices are defined as:

$$E_R = |t^s - g^s| + |t^e - g^e|$$
$$E_I = |\tau^s - g^s| + |\tau^e - g^e|$$

where $E_R$ and $E_I$ denote the corresponding boundary error of refined segment choices and initial segment choices. In Fig. 7, the horizontal and vertical coordinates respectively indicate the error range and the number of samples in the testing set. This figure shows the histograms of our choices before and after boundary refinement. Results indicate that our regression head changes the boundary error from the green curve to the red curve, which shows that our model reduces the boundary error by narrowing the decision space. More samples are distributed in the low error ranges after refinement, which leads to more accurate video grounding.

4) Effect of expansion ratio $\eta$: This section studies the impact of the extension ratio $\eta$ in defining start, center, and end regions. We evaluate 5 different values of $\eta$ from 0.1 to 0.5 in experiments. The results for various $\eta$ are shown in the Tab. VIII. It highlights that our extended boundary regions always contribute to performance improvement in video grounding. The extension ratio $\eta = 0.0$ indicates no extension for the boundary regions, namely no starting and ending areas, which is equivalent to the configuration without regression heads in Tab. VII. We take the TACoS and Charades-STA datasets for evaluation. From these results, we can find that the performance of our model first increases and then gradually decreases along with $\eta$ raises. The best results are achieved when $\eta = 0.2$ for a trade-off between evaluation metrics. However, all the results are better than the counterpart experiment without extension, demonstrating the effectiveness and robustness of our triple-span representation.
5) Effect of expand ratio $\phi$ in the suppression branch: This section studies the impact of the extension ratio $\phi$ in the supervision of suppression branch. When $\phi = 0.5$, we have doubled the annotated ground-truth segment. If we still continue to expand the region for supervision, the suppression branch will be affected by the noise from irrelevant video content. This is total different from our original intention of proposing the suppression branch. Thus, we only consider the value of $\phi$ varies in the range $[0, 0.5]$. The corresponding results are shown in Tab. IX. The evaluation is based on the TACoS and ActivityNet-Captions benchmarks in terms of Rank 1 at different IoU thresholds and mIoU. It can be observed that the overall change of the grounding performance is small. For the TACoS benchmark, our model first improves the performance from 27.73% to 28.39% as $\phi$ increases, and then drops from the highest point to 27.53%. The similar trend can be seen in the Charades-STA benchmark.

6) Effect of sampling number $N_a, N_c, N_e$: To confirm the effect of the RoI align operation, we conduct experiments to evaluate how different sampling locations affect the grounding performance in our framework. All the comparison are conducted on TACoS and ActivityNet-Captions (denoted as ANet for simplification) benchmarks. As shown in Tab. X, the experiment that sampling 8/16/8 from start, center, and end region respectively achieves the best performance. The 8/2/8 experiment means that mainly adopting the start and end local region features for boundary regression obtains the lowest performance. Compared with other four experiments, it indicates the importance of the center region features.

7) Visualization: For qualitative analysis of our model, we show two examples in Fig. 9(a) and 9(b) from ActivityNet-Captions and TACoS benchmarks, respectively. We denote the ground truth, initial, and refined choice with cyan, green, and red colors. From these two figures, we can see that the video span located by our grounding module is closer to the ground truth instance than that by initial choice. Compared with the initial best-matched choice, our grounding module refines its corresponding boundaries and makes it boundary sensitive for more accurate localization. For example, in the TACoS dataset, the target video span of the query "He twisted lime on juicer" is from 57.96s to 97.35s. The best-matched choice from initialization locates this query in the range from 58.92s to 99.70s. This grounding result is inaccurate, and its starting and ending timestamps should be earlier. Our model, especially the grounding module, accurately locates the boundary in 57.23s (starting) and 96.35s (ending).
V. CONCLUSION

In this paper, we formulate video grounding into a novel multi-choice video reading comprehension problem. In our solution, some video segments are predefined based on a temporal proposal map and then a deep model is used to select the best-matched segment choice among all candidates and then refine the boundaries of the selected one for accurate boundary localization. This solution leads to a consistent temporal proposal map and then a deep model is used to select solution, some video segments are predefined based on a

Acknowledgement. This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research through the Visual Computing Center (VCC) funding, the Beijing Natural Science Foundation (19L2040) and the National Natural Science Foundation of China (61772513). We also thank the support from CloudWalk Technology Co., Ltd.

REFERENCES


[61] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele,


