Parallel Pathway Dense Video Captioning With Deformable Transformer

WANGYU CHOI1, JIASI CHEN2, (Member, IEEE), AND JONGWON YOON1, (Member, IEEE)

1Department of Computer Science and Engineering (Major in Bio Artificial Intelligence), Hanyang University, Ansan 15588, South Korea
2Department of Computer Science and Engineering, University of California Riverside, Riverside, CA 92521, USA

Corresponding author: Jongwon Yoon (jongwon@hanyang.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of South Korea (NRF) funded by the Ministry of Education under Grant NRF-2022R1A2C1008743; and in part by the Ministry of Science and ICT (MSIT), South Korea, under the Grand Information and Communication Technology Research Center Support Program supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP), under Grant IITP-2020-101741.

ABSTRACT Dense video captioning is a very challenging task because it requires a high-level understanding of the video story, as well as pinpointing details such as objects and motions for a consistent and fluent description of the video. Many existing solutions divide this problem into two sub-tasks, event detection and captioning, and solve them sequentially ("localize-then-describe" or reverse). Consequently, the final outcome is highly dependent on the performance of the preceding modules. In this paper, we decompose this sequential approach by proposing a parallel pathway dense video captioning framework that localizes and describes events simultaneously without any bottlenecks. We introduce a representation organization network at the branching point of the parallel pathway to organize the encoded video feature by considering the entire storyline. Then, an event localizer focuses to localize events without any event proposal generation network, a sentence generator describes events while considering the fluency and coherency of sentences. Our method has several advantages over existing work: (i) the final output does not depend on the output of the preceding modules, (ii) it improves existing parallel decoding methods by relieving the bottleneck of information. We evaluate the performance of PPVC on large-scale benchmark datasets, the ActivityNet Captions, and YouCook2. PPVC not only outperforms existing algorithms on the majority of metrics but also improves on both datasets by 5.4% and 4.9% compared to the state-of-the-art parallel decoding method.

INDEX TERMS Machine learning, deep learning, video and language, video captioning.

I. INTRODUCTION

With the advent of large-scale video activity datasets [1], [2], [3], video understanding has become one of the hot topics in computer vision. Based on this interest, research on video understanding has successfully addressed various tasks, starting with activity classification and detection, and more complex video captioning [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. Videos naturally contain multiple events, such as one or more objects and motions appearing simultaneously at various lengths, however, the above-mentioned approaches provide limited information, e.g., only a sentence for a short trimmed video.

The associate editor coordinating the review of this manuscript and approving it for publication was Amin Zehtabian.

This lack of detail is not sufficient to describe the overall storyline of the video.

Recently, many methods for dense video captioning have been proposed for more detailed and rich video descriptions [19], [20], [21], [22], [23], [24], [25]. They can be classified into three classes from an architectural point of view: bottom-up, top-down, and parallelism. The bottom-up method detects the temporal regions (i.e., events) that are the core of the video story, and then generates a sentence describing each event. A top-down approach is to describe the entire video in a paragraph and localize each sentence to the timeline of the video. The parallel approach simultaneously decodes sentences and events from encoded video features.

A lot of work on various approaches has resulted in significant performance improvements, but there is still room
for improvement. Methods with a bottom-up or top-down sequential architecture suffer from the fact that the performance of subsequent modules is highly dependent on the preceding ones. No matter how sophisticated the captioning module is, it is exceedingly difficult to construct polite phrases for poorly defined events using the bottom-up strategy. For this reason, many sequential pipeline solutions ([19], [20], [21], [23], [24]) cannot be trained end-to-end and require relatively complex training techniques. In addition, the majority of sequential approaches ([19], [20], [21], [22], [23], [26]) generate many event proposals with the event proposal network and eliminate duplicate event proposals with handcrafted algorithms such as MNS. These algorithms require additional hyper-parameters (e.g., threshold) that greatly affect the outcome.

To address this problem, parallel decoding approaches have been proposed recently [25]. They improve performance by removing sequential dependencies; however, they introduce the risk of information bottlenecks at the branch point. The bottleneck limits performance by preventing the flow of information that requires localization and captioning. In other words, they still have an “encode-then-decode” procedure, so the bottleneck in the branch leads to poor results. For instance, the most recently proposed method [25], a parallel decoding approach with an event counter, is restricted to creating a large number of events and detailed descriptions for long video.

With careful analysis of the aforementioned challenges, we obtain the following inferences for high-quality dense video captioning. A parallel architecture could be employed to reduce the dependency between the captioning and localization modules while eliminating the branch’s bottleneck. Even though the parallel architecture helps get rid of dependencies between modules, branch points still need to be carefully designed to avoid bottlenecks.

This work proposes PPVC, a parallel pathway dense video captioning that breaks sequential processing and mitigates the negative impacts of preceding modules on succeeding modules. Unlike the bottom-up or top-down pipeline adopted by most existing methods, our proposed framework performs captioning and localization at the same time as depicted in Figure 1. Moreover, to mitigate the bottleneck of information, a challenge of the existing parallel decoding approach, we pass all potential information at branch points, filter out unnecessary things just before decoding, and introduce a multi-stack cross-attention mechanism to exploit multiple information for localization and capturing.

We evaluate the performance of PPVC on the ActivityNet Captions datasets [19]. PPVC achieves superior performance objectively and subjectively compared with state-of-the-art solutions. Specifically, PPVC not only overcomes the inherent problems of the sequential paradigm but also outperforms existing bottom-up and top-down methods. Compared with the state-of-the-art parallel approaches, PPVC is the best in most metrics. Regarding the quality of the outcomes, PPVC provides a detailed and fluent explanation with an adequate number of events. The main contributions of this paper are summarized as follows.

- We propose a parallel pathway dense video framework, PPVC by alleviating the bottleneck at the branch point.
- PPVC provides richer/fluent descriptions compared to the state-of-the-art parallel approaches and outperforms the other methods in most metrics.

The rest of this paper is organized as follows. In Section II, we discuss the related work on the dense video captioning task and differentiate our work from the state-of-the-art. We propose PPVC, a parallel pathway dense video captioning method in Section III. Section V evaluates the performance of PPVC in detail. We conclude the paper in Section VI.

II. RELATED WORK AND MOTIVATION

A. VIDEO CAPTIONING

Recently, frameworks for video-to-language tasks [4], [13] have been proposed. They are motivated by sequence-to-sequence tasks (e.g., machine translation [27], [28]) and image captioning [29], [30], [31], [32], [33], [34], and have adopted architectures such as recurrent neural networks (RNNs), long short-term memories (LSTMs), and gated recurrent units (GRUs). In addition, convolutional neural networks (CNNs) are used to extract features of visual information from the video. For example, basic frameworks [4], [13] capture visual features from a video using CNNs, then encode the video features and decode them into natural language using LSTMs. To improve the captioning quality, subsequent work has extended the basic encoder-decoder structure by incorporating a temporal attention mechanism [5], hierarchical RNNs [6], [8], LSTMs with visual-semantic embedding [7], a semantic decoder [9], [10], [11], reconstruction network [12], spatial-hard attention [14], and reinforced adaptive attention [15]. Despite a lot of effort, they are limited to generating a single sentence from a video. To overcome this limitation, several works [35], [36], [37] have emerged that fluently describe a video in one paragraph.
**B. DENSE VIDEO CAPTIONING**

In dense video captioning, it is required to detect multiple overlapping events in the video and generate appropriate sentences for each. Many dense video captioning techniques [19], [20], [21], [23] attempt to solve two sub-tasks (i.e., event detection and then event captioning) sequentially. Krishna et al. [19] first select events with the event proposal module configured with LSTM, and then caption each event using the captioning module with the attention mechanism. In [20], event classification with temporal coordinate and descriptiveness regression is employed to improve the performance of the event detection module. Wang et al. [21] adopt a bidirectional LSTM to improve the quality of the event proposal module. Zhou et al. [22] proposed an end-to-end trainable framework based on Transformer [38], attempting to bridge the two sub-problems (i.e., event detection and captioning). The authors in [23] point out that existing approaches generate events independently and adopt single-stream temporal action proposals [39] in the event proposal module to consider temporal dependencies between the events. For more accurate and rich captioning, the authors in [40] proposed a bi-modal transformer that utilizes multi-modal inputs (i.e., audio and visual information). Most previous approaches follow a bottom-up strategy (i.e., detect-then-describe), and if event proposals are miss-defined, captioning will also be affected. To this end, Deng et al. [24] proposed a top-down framework that does not propose events directly from the video. They create a video-level story (paragraph) from the video, then ground the story, and finally refine the sentences.

All of the above-mentioned methods adopt a sequential structure, which may cause a problem of affecting subsequent modules in a cascade manner due to the ill-defined preceding module. To address this, Wang et al. [25] proposed PDVC, a framework for localization and captioning in parallel by employing multiple heads in a transformer decoder. Although PDVC has achieved significant performance improvement over sequential architectures, it leaves the challenge of bottleneck problems at the branch point.

**C. MOTIVATION AND DESIGN CHOICE**

Inspired by Wang et al. [25], our framework also employs a parallel pathway pipeline. Parallel approaches such as PDVC [25] effectively overcome the limitations of sequential approaches. However, there is still room for improvement in terms of bottlenecks at the branching point. Even though the video contains many salient temporal regions, the low number of queries in the query-based deformable transformer can restrict the number of events (i.e., fluent descriptions). The authors of [25] proposed an event counter module to determine the number of queries, but the event counter usually underestimates the number of events (details in Figure 3). This hinders the video’s detailed explanation in a small number of sentences. Second, parallel decoding breaks the sequential structure of localize-then-describe, but the encoding-then-decoding procedure remains sequential. This implies that poorly defined features during encoding can negatively impact both localization and captioning.

To alleviate the bottleneck of the number of queries, at the branch point, we employ a large number of event queries to extract all potential events. We then deal with the problem of bottlenecks by filtering out unnecessary events just before decoding. Second, parallel decoding breaks the sequential structure of localize-then-describe, but the encoding-then-decoding procedure remains sequential. This implies that poorly defined features during encoding can negatively impact both localization and captioning. To mitigate this, we localize and describe the events by simultaneously referring to both organized and raw features using multi-stack cross-attention. Therefore, we design a framework that breaks the sequential flow, extracts the core information without query restrictions, and then localizes/describes the events in parallel based on rich input using multi-stack cross-attention.

**III. METHOD**

This section describes the core ideas and the network architecture of PPVC in detail.

**A. OVERVIEW**

For a given video, dense video captioning is the task of detecting multiple overlapping events and generating sentences accordingly. Existing methods generally divide the whole process into two sub-problems, event detection, and sentence generation, and solve them one by one in order. The performance of the two modules is tightly coupled, and this sequential processing method causes a major flaw: the result of the preceding module greatly affects the performance of the subsequent module. However, parallel architectures have the challenge that the bottleneck of information at branch points can limit the outcomes. Therefore, it is critical to decouple the dependency between the two modules by breaking the sequential processing paradigm while not occurring a bottleneck.

With this consideration, we construct two sub-problems of dense video captioning in parallel. We first organize the encoded video features into highly relevant ones to

---

**Algorithm 1 Parallel Pathway Dense Video Captioning**

1. **procedure** PPVC(V)
2. \( H_o \leftarrow E(V) \) \( \triangleright \) Encode video
3. **for** \( k = 1 \) to \( K \) **do**
4. \( H_o^{gated} \leftarrow O(H_o, q_k) \) \( \triangleright \) Organize features
5. \( H_o \leftarrow H_o \times G(H_o) \) \( \triangleright \) Filter out
6. \( s_k \leftarrow S(H_o^{gated}, H_o) \) \( \triangleright \) Describe
7. \( E_k \leftarrow L(H_o^{gated}, H_o) \) \( \triangleright \) Localize
8. **end for**
9. **return** \( \{E_k\}_{i=1}^{K}, \{s_i\}_{i=1}^{K} \)
10. **end procedure**
understand the core context of the video, rather than directly generating events and sentences from the video. PPVC consists of five modules: video encoder $E$, representation organizer $O$, event localizer $L$, sentence generator $S$, and gating network $G$ as depicted in Figure 2.

Given a video $V$, it is converted to a hidden state vector $H_v$ by the video encoder $E$. $H_v$ is fed into the representation organizer $O$ to organize the encoded video features that determine the context of the video storyline, which is the branch point of the parallel pathway without any bottleneck. The event localizer $L$ and sentence generator $S$ then localize the events $\{e_i\}_{i=1}^n$ and generate sentences $\{s_i\}_{i=1}^n$, respectively, while further mitigating bottlenecks using multi-stack cross attention. The gating network $G$ controls the flow at branch points and refines information by blocking the flow of unnecessary event queries. The procedure of PPVC is described in Algorithm 1.

**B. VIDEO ENCODER**

The goal of the video encoder $E$ is to extract the hidden state $H_v$ by encoding the spatiotemporal context of the video $V$. Our video encoder consists of a convolution network and a sequential data encoder. In particular, we adopt the C3D [41] network and the transformer encoder [38] with multi-head attention (MA) to understand the long-term temporal context of a video. First, the video encoder divides a given video $V$ into non-overlapping segments of fixed length (e.g., 8 frames). The C3D network takes a segment as input and extracts features $F_v \in \mathbb{R}^{T \times d_f}$, where $T$ is the number of segments and $d_f$ is the dimension of features. We apply a linear transformation to feed the video feature $F_v$ of dimension $d_f$ to the transformer encoder [38] of dimension $d_m$ as follows:

$$H_v^0 = \text{PE} (\text{Linear} (F_v))$$

(1)

Then, after applying the positional encoding (i.e., $\text{PE}()$) to $\text{Linear}(F_v)$, we start with the input $H_v^0$.

The transformer encoder, which contains multiple $N$ layers, is the core of our video encoder. Each layer is made up of self-attention and feed-forward modules, and at the end of each module, there is a residual connection and layer normalization. The transformer encoder outputs $H_v^{l+1}$ in one cycle given $H_v^l$, and the output of the last cycle—i.e., the last layer—is $H_v$. The following two equations can be used to describe the entire procedure:

$$H_v^{l+1} = \text{FFN} \left( \Psi \left( H_v^l + \text{MA} \left( H_v^l, H_v^l, H_v^l \right) \right) \right)$$

(2)

$$\text{FFN}(x) = \max (0, xW_1 + b_1)W_2 + b_2$$

(3)

$$\text{MA}(Q, K, V) = \text{Cat}(\text{ATT}(Q, K, V))_{i=1}^hW_O$$

(4)

$$\text{ATT}(Q, K, V) = \text{softmax} \left( \frac{(QW_Q)(KW_K)^T}{\sqrt{d_m}} \right) (VW_V)$$

(5)

where, $\Psi ()$ represents the layer normalization function. $\text{Cat}()$ denotes vector concatenation and $W_A$ are trainable parameters. We repeat the above process as many times as the number of encoder layers. Finally, the output of the video encoder is the output of the last layer $H_v$. 

![Architecture of the proposed method, PPVC.](image)
Specifically, the multi-head attention module contains the following two classes of trainable parameters: $W_{[K,Q,V]}$ for the query ($Q$), key ($K$), and value ($V$) attention matrix, and $W_O$, which is in charge of the output by concatenating attention heads. The attention head is calculated by scaled dot-product attention, as defined in Equation (5), with the same $H^l_o$ being served into $K$, $Q$, and $V$ for self-attention. The final output of multi-head attention is then produced by concatenating all attention heads and applying the weight of $W_O$. Two linear transformations and a ReLU [42] activation function make up the feed-forward network $FFN(\cdot)$. The first linear transformation (i.e., $W_1$) and ReLU activation are applied to the output $x$ of multi-head attention. Then it goes through a second linear transformation (i.e., $W_2$) with dropout as provided in Equation (3).

The representation organizer inherits the transformer’s learning principle. Specifically, with positional encoding and sequentially input video features, the transformer encoder learns the self-attention between features and organizes features for localization and captioning. Intuitively, each hidden vector in the outputs of the transformer encoder implicitly contains information for localization and captioning. Therefore, by training in an end-to-end manner from the input to the localization and captioning heads, the representation organizer is trained toward the optimal.

### C. REPRESENTATION ORGANIZATION

Before entering the parallel pathway, we introduce a representation organizer that organizes key encoded video features in the video’s spatiotemporal context and uses them as intermediate information for event localization and sentence generation. The goal of the representation organizer is to extract a representation of salient temporal regions in a video while implicitly producing all potential events. Specifically, given a video $V$, representation organizer outputs multiple representations $H_o$ considering the entire video story. Later, each representation serves as the core information for an event (i.e., timestamp and sentence), resulting in high-quality event localization and captioning.

For this, we adopt a query-based transformer decoder [43], [44] because of its high performance and efficiency in object detection tasks using transformers [38]. Given the $H_v$ obtained by the video encoder and the input organized representation vector $H^0_o$, the representation organizer computes the hidden state $H_o$ as follows:

$$H^l_o^\text{self} = \Psi \left( H^l_w + \text{MA} \left( H^l_o, H^l_o, H^l_o \right) \right)$$  \hspace{1cm} (6)

$$H^l_o^\text{cross} = \Psi \left( H^l_o^\text{self} + \text{MA} \left( H^l_o^\text{self}, H_v, H_v \right) \right)$$  \hspace{1cm} (7)

$$H^l_o^{l+1} = \Psi \left( H^l_o^\text{cross} + FFN \left( H^l_o^\text{cross} \right) \right)$$  \hspace{1cm} (8)

where, MA(·), FFN(·) and $\Psi$ (·) are the same as used in Equation 2 and Equation 3. $l$ is the sequence number of the transformer decoder. The output of the last layer is $H_o \in \mathbb{R}^{K \times d_{in}}$ ($K$ is the number of generated representations).

Here, as described above, we set the number of event queries to a fixed number of $K$ without depending on the event counter. The event counter is likely to output the average number regardless of the content based on the statistics of the entire dataset (Section V). We control the number of events by generating events for fixed $K$ event queries (i.e., the maximum number of events that PPVC can generate) and removing unnecessary ones.

### Choice of $K$

Determining $K$ requires great care as it limits the maximum number of events that PPVC can generate. With a high $K$, PPVC generates many events, so recall can be high, but precision is not guaranteed. On the other hand, a low $K$ increases precision, but does not guarantee high recall in videos containing many events. Therefore, we empirically set $K$ to 10 to measure the F1 score by varying $K$ to strike the balance between recall and precision (details in Section V-E).

### D. EVENT LOCALIZER

The goal of the event localizer is to predict the center and duration of an event in the video corresponding to an organized representation. Common problems with the event localizer are that it is difficult to predict how many events are included in the video, and the events must be determined by considering the correlations between events. Our event localization approach implicitly determines the number of events (i.e., the number of organized representations) in the previous step (representation organization), without an event counter or a proposal generation network. Therefore, this section only focuses on the localization of the timestamps corresponding to the organized representations in the entire video.

The event localizer is based on the transformer decoder [38]. The difference from the basic transformer decoder is that it does not have a self-attention block (i.e., not auto-regressive) and directly regresses the timestamp through several convolutional neural networks after the cross-attention block. For each organized representation $H_o$, we use a multi-head attention module to incorporate the entire video storyline (i.e., encoded video feature $H_v$) as follows:

$$H^\text{cross}_e = \Psi \left( H^\text{self}_e + \text{MA} \left( H^\text{self}_e, H_v, H_v \right) \right)$$  \hspace{1cm} (9)

We then apply several 1D convolutional layers on the temporal dimension and predict the ratio of center and duration timestamps to the entire video using two linear regression heads.

### E. SENTENCE GENERATOR

The goal of the sentence generator $S$, the second part of the parallel pathway, is to receive organized representations $\{H^l_o\}_{l=0}^N$ from the representation organizer and generate sentences $\{s^i\}_{i=0}^N$: $S \left( \{H^l_o\}_{l=0}^N \right) \rightarrow \{s^i\}_{i=0}^N$. One of the important points in sentence generation in a dense video captioning task is to maintain coherency. There are several approaches to this, but we guarantee coherency between sentences by directly incorporating the textual features of the
video. Directly merging video features instead of merging past sentences improves sentence coherency while avoiding referencing ill-defined words in the last sentence.

Our sentence generator is based on a transformer decoder [38], and most of the process is similar to \(O\). The difference from the basic transformer decoder is that there are two hidden states (i.e., two pairs of key and value) that are input to the cross-attention block: the visual hidden state \(H_v\) and the organized representation \(H_o\). Therefore, we design a multi-stack cross-attention module that refers to multiple hidden states. This applies the multi-head attention module sequentially to two or more hidden states by stacking, which makes it possible to utilize various information during decoding.

The sentence generation process using multi-stack cross-attention is as follows:

\[
H^\text{cross} = \psi \left( H^\text{self} \left( H^\text{cross} \right), H^\text{cross} \right) \tag{10}
\]

Note that we omit positional encoding, self-attention, and feed-forward processes because they are the same as the representation organizer (Section III-C).

F. GATING NETWORK
In addition to the modules described above, we employ a gating network that signals the end of a sequence of the representation organizer and controls the flow into parallel pathways. Since the representation organizer creates all of the possible organized representations, it is necessary to prevent the generation of redundant or unnecessary events by indicating the end of the sequence and limiting the flow of parallel pathways. In other words, the gating network consequently serves to increase precision in event localization. Our gating network outputs a confidence score for the organized representation through the sigmoid activation function after linear transformation as follows:

\[
g(x) = \sigma \left( \text{Linear}(x) \right) \tag{12}
\]

\[
H_o^\text{gated} = g(H_o) \cdot H_o \tag{13}
\]

It is simple but effective, prevents unnecessary representation generation, and as a result helps high-quality localization and captioning.

IV. TRAINING
PPVC is an end-to-end trainable model, and this section describes the loss functions for two output heads.

A. EVENT LOCALIZER
We use supervised learning for training the event localizer. The video encoder encodes a given video, then the representation organizer extracts the hidden state of the events according to the video story. Then, the event localizer outputs the center timestamps and duration of the event. Finally, combining the predicted events \(E = [e_c, e_d]\) and those of the ground-truth \(E^* = [e_c^*, e_d^*]\), we balanced logistic regression loss [45] as follows:

\[
L_C = \frac{1}{T} \sum_{t=1}^{T} \sum_{x \in \{c,d\}} (\alpha_t^+ e_{x,t}^* \log e_{x,t} + \alpha_t^- \log (1 - e_{x,t})) \tag{14}
\]

where, \(\alpha_t^+ = \frac{\sum_{e_{x,t}}}{\sum_{e_{x,t}}^+}\) and \(\alpha_t^- = \frac{T - \sum_{e_{x,t}}}{T - \sum_{e_{x,t}}^-}\) are the balance weights for positive and negative samples.

B. SENTENCE GENERATOR
We train a sentence generator by minimizing the negative log-likelihood of ground-truth words \([w_t^*]\). The loss function \(L_S\) is:

\[
L_S = -\log p(w_t^*|w_{<t}, H_v) \tag{15}
\]

where, \(j\) and \(w_{<t}^*\) are the current position in the sequence for prediction and the preceding ground-truth words, respectively. In addition, we apply label smoothing [46] for regularization of the model.

V. EXPERIMENTS
A. DATASET
We evaluate the performance of our framework on the two large-scale benchmark datasets, ActivityNet Captions [19] and YouCook2 [47]. The ActivityNet Captions dataset contains 19,994 Youtube videos, which are divided into three subsets: 10,009, 4,917, and 4,885 videos for training, validation, and testing, respectively. On average, the length of videos is 120 seconds, and each video has 3.65 pairs of events and sentences. Each sentence has, on average, 13.48 words. YouCook2 has 2,000 untrimmed videos of cooking activities with an average length of 320 seconds. The videos have 7.7 events and sentences on average. We use C3D [41] and TSN [48] features for fair comparison for the ActivityNet Captions dataset and the YouCook2 dataset, respectively. C3D features are obtained by passing non-overlapping, fixed-length video segments of 8 frames to the C3D network pre-trained on Sports-1M [1].

B. METRICS
We use publicly available evaluation code\(^1\) provided by the ActivityNet Captions Challenge. We measure recall and precision of event localization and METEOR [49], CIDEr [50], and BLEU [51] of sentences. Given a generated event and sentence pair, if its event has an overlapping larger than the threshold with any ground-truth events, the captioning score is computed by comparing the corresponding ground-truth sentence. Otherwise, the score is set to 0.

C. IMPLEMENTATION DETAILS
For all modules, the hidden size \(d_h\) of multi-head attention is 512, the number of attention heads is 8, and the encoder and decoder have 6 layers. The feed-forward network

\[^1\]https://github.com/ranjaykrishna/densevid_eval
used in the encoder and decoder is 2,048 dimensions. The residual and attention dropout ratio is 0.1. To prevent overfitting, we apply dropout to the visual input embedding layer. We train the model with adamW [52] for 30 epochs with a batch size of 1. We vary the learning rate as in [38] and set warmup steps to 10 epochs, which initially increases linearly from 0 to about 0.00005, and then decreases proportionally to the inverse square root. The label smoothing factor is set to 0.1.

### D. PERFORMANCE COMPARISON

We compare the performance of PPVC with several state-of-the-art methods [19], [20], [21], [22], [23], [24], [25], [26], [53], [54].

#### 1) EVENT LOCALIZATION

We first compare the performance of the event localizer among the two subtasks and present the results in Table 1. We evaluate the localization performance of PPVC with respect to the 4 temporal intersections of unions (@tIoU) thresholds on the Activity Captions validation set. PN stands for proposal network.

MFT, SDVC, and PDVC explicitly detect events through the event proposal networks. In contrast, our method uses the representation organizer to roughly compose representations of the events (i.e., implicit) and then localize each representation. Our event localizer achieves superior performance over MFT, SDVC, and PDVC in terms of F1 score. Specifically, PPVC surpasses MFT by a large margin, and although its precision is slightly lower than SDVC and PDVC, its recall is slightly superior, resulting in a higher F1 score. Based on these results, we can confirm the effectiveness of the representation organizer, the event localizer, and the gating network without any proposal network.

### TABLE 1. Performance comparison of the event localization.

| Method     | PN | Recall (@tIoU) | Training CE | Precision (@tIoU) | F1 |
|------------|----|----------------|-------------|-------------------|----|
|            |     | @0.3 @0.5 @0.7 @0.9 |            | @0.3 @0.5 @0.7 @0.9 |    |
| MFT 18 [36]| ✓  | 46.18 29.76 15.54 5.77 24.31 | ✓           | 86.34 68.79 38.30 12.19 51.41 |    |
| SDVC'19 [23]| ✓  | 93.41 76.40 42.40 10.10 55.58 | ✓           | 96.71 77.73 44.84 10.99 57.57 |    |
| PDVC'21 [25]| ✓  | 89.47 81.91 44.63 15.57 55.42 | ✓           | 97.16 78.09 42.68 14.40 58.07 |    |
| PPVC       | ✓  | 91.71 78.90 56.73 20.60 61.98 | ✓           | 96.23 73.80 37.66 12.61 55.07 |    |

#### 2) DENSE VIDEO CAPTIONING

Tables 2 and 3 show the comparisons of captioning performance for state-of-the-art methods using BLEU@N (B@N), CIDEr (C) and METEOR (M) on the ActivityNet validation set and YouCook2 dataset, respectively. Asterisk (*) indicates methods evaluated under an incomplete dataset (e.g., 80%) due to a download issue. CE and RL stand for cross-entropy and reinforcement learning, respectively. PPVC achieves superior performance compared to the state-of-the-art scheme in terms of the widely used METEOR metric in the ActivityNet Captions challenge.

A few methods [23], [24], [26] adopt reinforcement learning to further improve performance after cross-entropy training. They require extra-long training time for RL training and most importantly they tend to generate repetitive phrases [55]. The RL fine-tuning approach expects high metric values but yields rather poor results in terms of human readability. In short, metrics such as CIDEr and METEOR cannot perfectly evaluate human readability. In this paper, since we attach great importance to both quantitative and qualitative results, we do not adopt the RL training in the language model.

Note that, we exclude comparison methods that use anything other than C3D/TSN features (i.e., multi-modal) or train the captioning module with reinforcement learning after

### TABLE 2. Performance comparison of the dense video captioning (ActivityNet Captions validation set).

| Method     | Training CE | Feature | B@1 | B@2 | B@3 | B@4 | C | M   |
|------------|-------------|---------|-----|-----|-----|-----|---|-----|
| MFT 18'17 [19] | ✓           | ✓       | 10.81 | 4.57 | 1.90 | 0.71 | 12.43 | 5.69 |
| TDA-CG'18 [21] | ✓           | ✓       | -    | -   | -   | 1.31 | 7.99 | 5.86 |
| DVC'18 [20] | ✓           | ✓       | 12.22 | 5.72 | 2.27 | 0.73 | 12.61 | 6.93 |
| Efficient'20 [53] | ✓           | ✓       | -    | -   | 2.87 | 1.35 | 13.82 | 6.21 |
| ECHR'20 [26] | ✓           | ✓       | -    | -   | -   | 1.29 | 14.71 | 7.19 |
| WS-DEC'21 [54]* | ✓           | ✓       | 13.36 | 5.96 | 2.78 | 1.33 | 21.21 | 7.49 |
| PDVC'21 [25] | ✓           | ✓       | -    | -   | -   | 1.65 | 25.87 | 7.50 |
| PPVC       | ✓           | ✓       | 14.93 | 7.40 | 3.58 | 1.68 | 23.02 | 7.91 |

### TABLE 3. Performance comparison of the dense video captioning (YouCook2-TSN).

| Method     | B@4 | C | M   |
|------------|-----|---|-----|
| MFT 18 [22] | 0.30 | 6.10 | 3.18 |
| ECHR'18 [21] | -   | - | 3.82 |
| PDVC'21 [25] | 0.40 | 22.71 | 4.74 |
| SGR'21 [24] | -   | - | 4.35 |
| PPVC       | 0.89 | 19.70 | 4.94 |
TABLE 4. A performance comparison of the dense video captioning with ground-truth proposals.

| Method    | B@4 | C   | M   |
|-----------|-----|-----|-----|
| DCE’17 [19] | 1.60 | 8.88 | 25.12 |
| TDA-CG’18 [21] | 6.96 | -   | 25.24 |
| DVC’18 [20] | 1.62 | 10.33 | 25.24 |
| ECHR’20 [26] | 1.96 | 10.58 | 39.73 |
| PDVC’21 [25] | 2.64 | 10.54 | 47.26 |
| PPVC      | 2.76 | 10.48 | 49.31 |

cross-entropy training. RL adopts a natural language evaluation metric such as METEOR directly as a reward, which is very effective for high scores. However, it tends to generate repetitive phrases with long sentences [55], which reduces readability [21], [56]. In this work, we do not use the RL training in the language model since we place a high value on both quantitative and qualitative results.

We evaluate the PPVC with ground-truth proposals, examining only the captioning performance separately from event localization on the ActivityNet validation set, as shown in Table 4. Compared to other algorithms, PPVC is slightly inferior in CIDEr but superior in BLEU@4 and METEOR metrics, indicating that the events generated by PPVC are more accurate. We also compare the performance of PPVC with the state-of-the-art methods on the YouCook2 dataset, as shown in Table 3. Similarly, PPVC outperforms other algorithms in terms of METEOR and BLEU@4. Based on these results, we can see that PPVC is effective with parallel pathways instead of sequential paradigms. Compared to PDVC, which is a similar parallel decoding approach, PPVC achieves slightly better results in terms of BLEU and METEOR, indicating the effectiveness of representation organizers over event counter head.

We analyze PPVC in depth based on the evaluation mechanisms and values of the metrics. CIDEr (Consensus-based Image Description Evaluation) is a TF-IDF (Term Frequency-Inverse Document Frequency)-based natural language evaluation metric. By taking into account both the frequency of the words and the length of the document, TF-IDF determines the importance of each word. CIDEr evaluates a candidate sentence by vectorizing the TF-IDF values of words in the candidate sentence and reference sentences and comparing their similarity. Intuitively, it tends to ignore grammar and word order and concentrate more on event-specific words than common words in reference sentences. METEOR (Metric for Evaluation of Translation with Explicit ORdering), on the other hand, evaluates a candidate sentence using a penalty function for inconsistencies and weighted F-score (i.e., both recall and precision). It takes into account the portion of the generated sentence where the n-grams match those of the reference sentences. Intuitively, it tends to ignore grammar and word order and concentrate more on event-specific words than common words in reference sentences. Based on these facts, that PPVC has advantages in terms of sentence completeness using common words and drawbacks in generating event-specific words. The limitations of PPVC and future work are discussed in detail in Section VI.

TABLE 5. Average number of events per video.

| Method    | Approach | Num. events |
|-----------|----------|-------------|
| SDVC’19 [23] | PN  | 2.85 |
| PDVC’21 [25] | RC  | 3.03 |
| PPVC      | -        | 5.54 |

3) REPRESENTATION ORGANIZATION

For a more in-depth analysis of representation organizers, we examine the events generated from all videos included in the ActivityNet Captions validation set. Table 5 and Figure 3 show the number of events generated from each method,
FIGURE 6. Examples of qualitative results on ActivityNet Captions validation set. Sentences corresponding to the event are matched with the same color.

and the distribution of the number of events generated for all videos, respectively. In Table 5, PN, NMS, and EC are proposal network, non-maximum suppression, and event counter, respectively.

In PPVC, the number of events is widely distributed from 3 to 9, whereas PDVC generates only 3 to 5 events per video. In terms of the average number, PPVC, PDVC, and SDVC generate 5.54, 3.03, and 2.85 events, respectively, indicating that PPVC generates more sentences and a more detailed description of the video.

Furthermore, based on the above results, the event counter of PDVC outputs 3 (80.4%), which is the average of the number of events of the videos in the entire dataset. On the other hand, PPVC without an event counter generates proposals with a more diverse number of events.

E. ABLATION STUDY

1) ABLATION STUDY ON EACH MODULE

We conduct several ablation studies to precisely quantify the contribution of the core modules of PPVC. We compare the performance of 4 different models with different combinations of modules as follows: (i) vanilla PPVC: has a parallel pathway but no representation organizer and gating network, (ii) PPVC without representation organizer: directly localizes and describes the events from the encoded video, (iii) PPVC without gating network: does not control the flow to the event localizer and the sentence generator, (iv) PPVC (full model).

Figure 4 and Table 6 show the average recall and precision (i.e., localization) of an ablation study on the ActivityNet validation set and we have the following observations. “RO” and “GN” stand for the representation organizer and the gating network, respectively. First, vanilla PPVC eliminates the dependency between the two modules with a parallel

| Method   | Recall | Precision |
|----------|--------|-----------|
| Vanilla  | 44.53  | 32.92     |
| W/o RO   | 77.90  | 34.57     |
| W/o GN   | 74.37  | 39.93     |
| Full     | 61.98  | 58.33     |
architecture, however, suffers from localizing and describing events directly from the vast amounts of video features. We argue that the simple parallel pathway dense video captioning model may not be as effective as the conventional sequential method. Second, applying the representation organizer to vanilla PPVC, it can be seen that there is a significant performance gain. This result proves that the representation organizer acts as a filter to extract only the necessary (core) information and guides the event localizer and sentence generator. Third, the gating network is essential in PPVC because it controls the flow from the representation organizer to the parallel pathway and filters the unnecessary representations.

2) ABLATION STUDY ON THE NUMBER OF QUERIES
We train several times to determine the number of fixed queries, $K$, and measure F1 for the generated events by varying $K$. Based on the fact that the ActivityNet Captions dataset contains videos of up to 27 events, we examine the results for $K$ of 3, 5, 7, 10, 15, 20, 25, and 30. It can be seen that the best performance is when the number of queries is 10, as shown in Figure 5. The overall performance tends to improve as the number of queries rises, but when it exceeds 10, it starts to somewhat decline. PPVC has the following trade-offs, depending on the number of queries: Low query numbers cause PPVC to have a low recall in event localization; on the other hand, high query numbers cause a high recall but poor precision. Consequently, we set $K$ to 10 for the number of queries, which is the best F1 score (i.e., the harmonic mean of the recall and precision).

In short, by eliminating bottlenecks caused by the representation organizer and the gating network, PPVC achieves performance improvements over PDVC, an existing parallel decoding method. Figure 4 demonstrates the efficiency of the representation organizer and the gating network, while Figure 5 demonstrates the need for a larger number of queries.

F. QUALITATIVE RESULTS
Figure 6 illustrates the qualitative results of localizing and describing events in the video. First, it can be seen that PPVC localizes and describes videos with a large number of events (i.e., more than 4) in both examples. To detect more events, PPVC uses the representation organizer and the gating network, while Figure 5 demonstrates the need for a larger number of queries.

VI. CONCLUSION
In this paper, we propose a parallel pathway dense video captioning, that first organizes the encoded video features and then simultaneously performs event localization and sentence generation. The representation organizer arranges core information from the video by generating representations constituting the storyline. With the help of the representation organizer, the event localizer focuses on accurately predicting timestamps without generating any proposals. The sentence generator creates sentences for organized representations. In this way, we eliminate the dependency between modules, which is a disadvantage of the existing sequential pipeline. Furthermore, we alleviate the bottleneck problem at branch points beyond the existing method of parallel architecture with maximum event query count and multi-stack cross attention. Experiment results demonstrate that PPVC provides state-of-the-art performance (i.e., an improvement of 5.4% and 4.9%) on the ActivityNet Captions and YouCook2 datasets, respectively.

Both parallel pathway and bottleneck mitigation provide considerable performance gains, but there is still room for improvement. First, for localization, even though PPVC identifies more events than previous approaches, it still falls short of ground-truth events. Second, PPVC has a simple architecture for captioning compared to previous approaches, and its benefit is negligible compared to localization. In future work, we will thus continue to enhance the quality of parallel pathway dense video captioning while overcoming these two major limitations of PPVC.

REFERENCES
[1] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1725–1732.
[2] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, “ActivityNet: A large-scale video benchmark for human activity understanding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 961–970.
[3] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, “The kinetics human action video dataset,” 2017, arXiv:1705.06950.
[4] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, “Translating videos to natural language using deep recurrent neural networks,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., 2015, pp. 1494–1504.
[5] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, “Describing videos by exploiting temporal structure,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 4507–4515.
[6] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, “Hierarchical recurrent neural encoder for video representation with application to captioning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1029–1038.
[7] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, “Jointly modeling embedding and translation to bridge video and language,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4594–4602.
[8] L. Baraldi, C. Crana, and R. Cucchiara, “Hierarchical boundary-aware neural encoder for video captioning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1657–1666.
[9] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng, “Semantic compositional networks for visual captioning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5630–5639.
K. Cho, B. Van Merrienboer, C. C. Gülçehre, D. Bahdanau, F. Bougares, T. Wang, H. Zheng, M. Yu, Q. Tian, and H. Hu, “Event-centric hierarchical video captioning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3165–3173.

B. Wang, L. Ma, W. Liu, and T. Mei, “Reconstruction network for video captioning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7622–7631.

S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, “Sequence to sequence—Video to text,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 4534–4542.

A. Liu, Y. Qiu, Y. Wong, Y. Su, and T. Mei, “Attention is all you need,” in Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2017, pp. 6000–6010.

Y. Li, Y. Pan, T. Yao, and T. Mei, “Comprehending and ordering semantics for image captioning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 17990–17999.

H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, “Video paragraph captioning using hierarchical recurrent neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4584–4593.

Y. Xiong, B. Dai, and D. Lin, “Move forward and tell: A progressive generator of video descriptions,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 468–483.

J. Lei, L. Wang, Y. Shen, D. Yu, T. Berg, and M. Bansal, “MART: Memory-augmented recurrent transformer for coherent video paragraph captioning,” in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, 2020, pp. 2603–2614.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, “XLNet: A general-language pre-trained model for translation and other NLP tasks,” in Proc. 56th Annu. Meeting Assoc. Comput. Linguistics, 2019, pp. 4108–4126.

Y. Yu, H. Ko, J. Choi, and G. Kim, “End-to-end concept word detection for video captioning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2740–2755.

H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in Proc. Conf. Empir. Methods Natural Lang. Proc., 2014, pp. 1724–1734.

S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, “Sequence to sequence—Video to text,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 4534–4542.

A. Liu, Y. Qiu, Y. Wong, Y. Su, and T. Mei, “Attention is all you need,” in Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2017, pp. 6000–6010.
WANGYU CHOI received the B.S. degree in computer science and engineering from the Tech University of Korea, in 2017. He is currently pursuing the Ph.D. degree in computer science and engineering with Hanyang University, South Korea. His research interests include the video streaming, mobile networking, and computer vision.

JIASI CHEN (Member, IEEE) received the B.S. degree from Columbia University, New York, NY, USA, with internships at the AT&T Laboratories Research, Florham Park, NJ, USA, and the NEC Laboratories America, Princeton, NJ, USA, and the Ph.D. degree from Princeton University, Princeton. She is currently an Associate Professor with the Department of Computer Science and Engineering, University of California at Riverside, Riverside, CA, USA. Her current research interests include edge computing, wireless and mobile systems, multimedia networking, and machine learning at the network edge to aid augmented reality (AR)/virtual reality (VR) applications. She was a recipient of the Hellman Fellowship and the UCR Regents Faculty Fellowship.

JONGWON YOON (Member, IEEE) received the B.S. degree in computer science from Korea University, in 2007, and the M.S. and Ph.D. degrees from the University of Wisconsin–Madison, in 2012 and 2014, respectively. He is currently an Associate Professor with the Department of Computer Science and Engineering, Hanyang University, South Korea. He is the Founding Director of the Intelligent Machines Laboratory which broadly focuses on research in machine learning, artificial intelligence, mobile systems, and HCI. He was a recipient of the Lawrence Landweber Fellowship during the Ph.D. study.