

# Live Video Captioning

Eduardo Blanco-Fernández<sup>1</sup>, Carlos Gutiérrez-Álvarez<sup>1</sup>,  
Nadia Nasri<sup>2</sup>, Saturnino Maldonado-Bascón<sup>1</sup>,  
Roberto J. López-Sastre<sup>1\*</sup>

<sup>1</sup>Department of Signal Theory and Communications, University of  
Alcalá, Alcalá de Henares, 28805, Spain.

<sup>2</sup>Department of Computer Science and Artificial Intelligence, University  
of Alicante, San Vicente del Raspeig, 03690, Spain.

\*Corresponding author(s). E-mail(s): [robertoj.lopez@uah.es](mailto:robertoj.lopez@uah.es);  
Contributing authors: [e.blancof@edu.uah.es](mailto:e.blancof@edu.uah.es); [carlos.gutierrezalva@uah.es](mailto:carlos.gutierrezalva@uah.es);  
[nadia.nasri@ua.es](mailto:nadia.nasri@ua.es); [saturnino.maldonado@uah.es](mailto:saturnino.maldonado@uah.es);

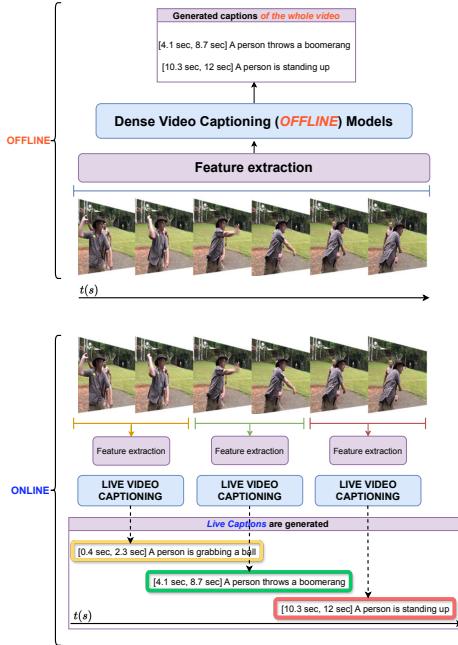
## Abstract

Dense video captioning involves detecting and describing events within video sequences. Traditional methods operate in an offline setting, assuming the entire video is available for analysis. In contrast, in this work we introduce a groundbreaking paradigm: Live Video Captioning (LVC), where captions must be generated for video streams in an online manner. This shift brings unique challenges, including processing partial observations of the events and the need for a temporal anticipation of the actions.

We formally define the novel problem of LVC and propose innovative evaluation metrics specifically designed for this online scenario, demonstrating their advantages over traditional metrics. To address the novel complexities of LVC, we present a new model that combines deformable transformers with temporal filtering, enabling effective captioning over video streams.

Extensive experiments on the ActivityNet Captions dataset validate the proposed approach, showcasing its superior performance in the LVC setting compared to state-of-the-art offline methods. To foster further research, we provide the results of our model and an evaluation toolkit with the new metrics integrated at: <https://github.com/gramuah/lvc>.

**Keywords:** dense video captioning; online video analysis; transformers; deep learning; computer vision



**Fig. 1** Above: Traditional models of dense video captioning work offline, accessing the whole video to generate the captions. Down: The live video captioning models must generate the captions for the video stream, in an online manner, and working with partial observations of the video.

## 1 Introduction

As a growing field within video understanding, video captioning has gathered significant attention recently [1–5]. The goal of these video captioning models is to produce a natural sentence that encapsulates the primary event in a *short* video. These models utilize datasets tailored to the described problem (e.g., MSR-VTT [6], VATEX [7]), where short video segments and their corresponding annotations in the form of captions are provided. Nevertheless, because real-world videos are often lengthy, untrimmed, and feature multiple simultaneous events alongside background content, the aforementioned single-sentence video captioning models typically produce lackluster and less informative descriptions. To tackle this more complex scenario, captioning approaches must both locate and describe the events occurring in long videos; this problem is known as *dense* video captioning [1, 8–12].

The majority of real-life videos encompass numerous events that may unfold simultaneously. For instance, in a video featuring “a waiter carrying food to a table” one may also observe another “individual eating and drinking” or a “woman sitting down”. Dense video captioning models must generate descriptions for each event unfolding in a video, precisely indicating the start and end times of each event.

However, the current state-of-the-art in dense video captioning primarily offers *offline* solutions. As illustrated in the top part of Figure 1, these models assume

access to the entire video before generating captions. In contrast, our work introduces a paradigm shift termed Live Video Captioning (LVC), depicted in the bottom part of Figure 1. In LVC, the challenge is to generate dense captions operating in an *online* manner. This introduces significant new challenges for traditional offline models. Firstly, in an LVC scenario, it is not feasible to access the entire video to generate dense captions. LVC models must work with *partial* video observations, ideally through video streams, continuously generating dense captions as events unfold. This restriction is particularly detrimental to models requiring an action proposal generation phase, e.g. [11, 13, 14]. Secondly, LVC models must be capable of anticipating actions. Working with partial observations limits the semantic information available compared to traditional offline models, making event identification more challenging. Lastly, LVC solutions need to incorporate temporal attention and filtering mechanisms to refine their caption predictions as the video streams evolve.

From an applicability standpoint, the LVC problem introduces new scenarios where live captioning is crucial, and traditional offline models would fail to perform effectively. LVC solutions must strike a balance between precision and speed, as both are equally vital. The utility of LVC models is substantial, enabling a wide range of applications, from content summarization and accessibility support in live broadcasts to online perception and decision-making in robotics and video surveillance. While the former paves the way for novel, accessible, and interactive multimedia experiences, the latter will be crucial for the next generation of autonomous navigation and human-robot interaction systems. These are, without question, scenarios where traditional offline video captioning models fall short.

Interestingly, to the best of our knowledge, the problem of Live Video Captioning (LVC) has not been extensively investigated. In this work, we make several key contributions. First, we offer a formal description of the LVC problem, detailed in Section 3.1. Second, we introduce a novel model for LVC that combines deformable transformers with a temporal filtering mechanism to generate dense captions over video streams, as described in Section 3.2. Third, we propose new evaluation metrics specifically designed for the online scenario, demonstrating that traditional offline metrics are inadequate, as discussed in Section 3.3. Finally, in Section 4, we provide a comprehensive experimental evaluation using the ActivityNet Captions dataset, showcasing both the performance of our new LVC model and the effectiveness of the new metrics. We also include a comparison with state-of-the-art offline methods, highlighting their limitations in addressing the new online scenario.

## 2 Related work

**Dense video captioning** presents a multifaceted challenge, intertwining event localization and event captioning. Krishna et al. [13] introduced the inaugural dense video captioning model, incorporating a multi-scale proposal module for localization alongside an attention-based LSTM for contextually informed caption generation. Subsequent research endeavors have aimed to enhance event representations through various means, including context modeling [15, 16], event-level relationships [12], and

multi-modal feature fusion [9, 17], thus facilitating more precise and informative caption generation.

Previous methods have struggled with integrating the localization and captioning modules effectively. Attempts to address this issue have led researchers to explore interactions between the two sub-tasks. Li et al. [18] propose a proxy task, predicting language rewards of generated sentences, to enhance the optimization of the localization module. Similarly, Zhou et al. [19] introduce a differential masking mechanism, linking the gradient flow from captioning loss to proposals’ boundaries, thereby facilitating joint optimization. In [10], the proposed approach exploits inter-task interactions by ensuring both sub-tasks share the same intermediate features. Furthermore, the method employs a one-to-one matching strategy between intermediate feature vectors and target event instances, resulting in discriminative features for captioning.

All the aforementioned methods share an important feature: they tackle the problem using pipelines designed to operate *offline*. In other words, the results of all these models are optimal when they have access to the entire video for which they generate the dense captions.

We propose in this work an approach that addresses the problem of dense video captioning in an *online* fashion. This new problem is named as live video captioning (LVC). Ideally, in the LVC problem, the captions must be generated as soon as possible, by processing the video stream. This means that the models need to be adapted to work with partial observations of the video content, and, under this condition, produce dense captions as accurate as possible. Note that other online approaches have been explored, for example, in the different problem of action detection (e. g. [20–26]).

For the dense video captioning problem, only, to the best of our knowledge, Hori et al. [27] have proposed a multi-modal captioning approach that uses a timing detector so that the captions can be generated in the early stages of an event-triggered video clip. This problem can be termed as early video captioning, where the target consists in evaluating the latency ratio needed to reach the same performance of an offline video captioning model for an event-triggered video. Similar simplified experimental setups where explored in the context of early event detection in video, e.g. [28–30]. Note that these problems are different from our live video captioning. We claim these simplified setups are not representative for practical applications, where occurrences of possibly many different actions need to be detected and a correct caption generated in an online manner, in long video recordings with widely varying content. When it comes to a live video captioning system, the model should be continuously processing the video stream, and, when necessary, producing dense video captions. This necessitates precisely recognizing the current action at any point of its development. Furthermore, to complete the LVC task, one must distinguish the action from a range of negative inputs, such as background frames in which no pertinent actions are occurring.

Overall, we propose a live video captioning model that parallelizes localization, selection, and captioning tasks within a single end-to-end model, based on deformable transformers [31, 32], simplifying the process while ensuring the online generation of accurate and coherent captions. Specifically, our approach is based on the deformable transformer of Zhu et al. [32], which is a model that was introduced as an architecture to improve the performance of object detectors by attending to sparse spatial

Work	Joint Loc. & Cap.	LVC	Contributions
[13]	x	x	First work; Proposals module + attention-based LSTM for caption generation.
[15, 16]	x	x	Enriched event representations by context modeling.
[12]	x	x	Incorporate event-level relationship.
[9, 17]	x	x	Use of multi-modal feature fusion.
[18]	✓	x	Use of proxy task based on language rewards to enhance the localization.
[10]	✓	x	Inter-task interactions with deformable transformers.
[19]	✓	x	Differential masking for linking the gradient flow from captioning loss to proposals' boundaries.
LVC (Ours)	✓	✓	Deformable transformers + temporal filtering for video streams.

**Table 1** Review of other models for the problem of dense video captioning. We organize the literature based on whether the models address the problems of localization and captioning jointly, and whether they can handle the proposed LVC scenario or operate offline. We also summarize some of the main contributions of the works.

locations and incorporating multi-scale feature representations. We have adapted this deformable transformer architecture to process video sequences for the novel problem of LVC. Our localization and captioning modules process the streaming video, producing online dense captions that are enhanced with an extra filtering process.

Similar transformers-based architectures were also used in offline dense video captioning models, e.g. [9, 10]. For instance, while both our model and [10] leverage deformable transformers, their architectural designs reflect fundamentally different objectives due to the contrasting demands of their respective tasks. In general, the main differences with respect the previous offline dense video captioning models [9, 10], that also use transformer-based architectures are as follows. First, they assume full access to the entire video sequence during both training and inference. This enables the models to perform holistic event localization and captioning by leveraging global temporal information. In contrast, our LVC model processes partial, sequentially available video segments under causal constraints, where no future frames are accessible. To accommodate this, our deformable transformer is modified to focus solely on the features within the current segment and refine predictions dynamically as new frames arrive. Second, in the offline dense video captioning setting, a multi-scale deformable attention can be used to process densely extracted frame features from the entire video. This attention mechanism ensures that the model captures relationships between temporally distant events. In our LVC model, we adapt the multi-scale deformable attention mechanism to operate on temporally truncated input streams. For instance, our event queries are tailored for online processing, designed to anticipate and refine event boundaries dynamically as frames are processed. This iterative refinement aligns with the causal nature of the LVC task, allowing the model to balance precision with latency. We also add a temporal filtering mechanism, integrated with the learnable event queries, to dynamically generate the dense captions for the events within the limited temporal context available, enabling precise and actionable online predictions. These distinctions highlight how our use of deformable transformers is uniquely adapted to the challenges of LVC.

In table 1 we provide a global comparison of the main models for dense video captioning. Note that only our approach has been specifically designed for the novel LVC problem.

### 3 Live Video Captioning

#### 3.1 Live Video Captioning: problem formulation

We define the problem of Live Video Captioning (LVC) as the process of obtaining dense captions for a video stream as soon as the video frames are available. Unlike the traditional video captioning scenario, which we refer to as offline video captioning, we do not have access to the entire video for analysis. Instead, LVC models can only access the content coming from a video stream up to the time instant  $t$  to generate caption predictions for that instant. In other words, LVC models process the video in an online fashion, implying that the dense caption generation system has access to the current information of the video and past information, but never future information. Therefore, these systems are inherently causal.

LVC models face unique challenges when compared to traditional offline captioning systems. In offline systems, the model has access to the entire video, allowing for accurate temporal segmentation and caption generation after full observation. In contrast, LVC models must generate captions in an online fashion, working with partial, often incomplete frames, and make predictions with limited information. This online constraint introduces several challenges: (i) accurate temporal segmentation with minimal context, (ii) the need for anticipation of actions before they are fully visible, and (iii) ensuring that caption generation remains both fast and precise.

We must assume that for LVC solutions, there is always information yet to be revealed. Caption predictions are made based on partial content of the video, in case it is already recorded. However, it is in the context of live video streams where LVC models gain special relevance. We can think of the following applications and problems. For instance, consider a surveillance camera monitoring a busy public space, where the flow of events is unpredictable. In such a scenario, we can never be certain about what might occur next—a sudden altercation, a person leaving an unattended bag, or an individual entering a restricted area. However, it is crucial to generate dense captions as soon as the visual information is available to the system. This capability enables timely detection and response to critical events, enhancing safety and situational awareness. Another illustrative example is a robot equipped with a camera, tasked with generating detailed descriptions of the scenes it encounters while navigating a dynamic environment, such as a bustling factory floor or a crowded shopping mall. The robot might describe activities like workers assembling parts, customers interacting with products, or obstacles in its path. This continuous scene interpretation is vital for tasks like human-robot interaction, navigation, and situational understanding. Any traditional offline model for dense video captioning would not be able to generate these descriptions, as these models have been trained to operate only when the entire video is available. For instance, models that rely on action proposals (e. g. [13, 15, 16]) would be severely affected. In the novel LVC paradigm, it is not desirable to wait until the action is finished to have a correct caption prediction for it.

Therefore, a critical advantage of Live Video Captioning (LVC) is its ability to anticipate actions and provide temporally precise localization of events, a limitation commonly observed in offline dense video captioning methods. For instance, in a video where a person begins to raise a glass to drink, followed by a second person setting

a plate on the table, offline methods would generate overlapping captions like '[2.0s-6.0s] A person moves' and '[5.0s-9.0s] A person interacts with an object,' failing to distinguish between the two actions clearly or provide precise temporal boundaries. By contrast, LVC models would produce '[2.0s-4.0s] A person raises a glass' and '[4.0s-6.0s] Another person places a plate on the table,' with non-overlapping, accurate temporal segmentation. This novel scenario not only allows for early recognition of ongoing events but also improves temporal granularity, enabling a clearer distinction between simultaneous or sequential actions. These capabilities are vital for applications requiring immediate and accurate action detection, such as the monitoring systems or interactive robotics described.

In the new context of LVC, we proceed to define the fundamental properties that characterize it, which are as follows:

1. **Input assumption:** Streaming videos are assumed to be the natural inputs for LVC approaches, where neither length nor content of the *entire* video are accessible.
2. **Timeliness:** Captions must be generated as soon as the actions unfold.
3. **Causality:** Dense caption generation must be causal, so future frames cannot be used.
4. **Temporal adjustment:** The caption prediction must be adjusted to the temporal information available up to the time corresponding to the prediction instant.
5. **Irreversibility:** No post-processing or subsequent thresholding of caption scores can be applied once they are generated for a previous instant of time. LVC methods cannot revise past generated captions.

### 3.2 Our Live Video Captioning Model

For the implementation of our LVC model, we drew inspiration from the latest advances in solutions for offline dense video captioning [9, 10], where transformer-based architectures [31] were used.

As it is shown in Figure 2, we develop a deformable transformer model applied to the novel online dense video captioning problem. The deformable transformer model was introduced in [32] as an architecture to improve the performance of object detectors by attending to sparse spatial locations and incorporating multi-scale feature representations. Technically, our LVC model integrates the deformable transformer-based architecture to construct online caption predictions, taking temporal video segments of length  $\Delta t$  as inputs. Given an input video stream  $V_i = \{I_1, I_2, I_3, \dots\}$ , we first split it in video segments  $S_i$  of duration  $\Delta t$ , hence  $V_i = \{S_1, S_2, S_3, \dots\}$ . Note that our LVC does not have to access the entire video, as required for offline systems such as [9–11, 33].

These video snippets  $S_i$  are the inputs for a deformable transformer model, with the corresponding encoder and decoder. The following operational scheme is followed from the introduction of the video segment to the model until the captions are obtained. First, our model extracts the features for each of the frames in the video segment  $S_i$ .

For our goal of generating dense captions for the actions and events occurring in videos, we have chosen to use a pre-trained model for action recognition. In particular, for our experiments, we used the Temporally-Sensitive Pretraining (TSP) feature



**Fig. 2** LVC adopts a deformable *transformer*-based architecture to learn the interaction of different frames of the video, including learnable event queries to capture the significance of the relationship between frames and events. Two prediction heads run in parallel on the query features, leveraging mutual benefits between the two tasks and improving their performance together.

extractor [34]. However, our LVC model can be integrated with any action recognition feature extractor. We then rescale the temporal dimension of the feature maps obtained to a fixed length  $T$  using interpolation. To effectively use multi-scale features for forecasting events at different scales, we incorporate  $L$  temporal convolutional layers with a stride of 2 and a kernel size of 3. This approach generates feature sequences at various resolutions, ranging from  $T$  to  $\frac{T}{2^L}$ . These multi-scale features  $\{\mathbf{x}^l\}_{l=1}^L$ , combined with their positional embeddings, are input into the deformable transformer encoder, which captures frame-to-frame relationships across multiple scales.

Note that, accurate captioning for videos with multiple simultaneous events relies heavily on temporal information to segment overlapping actions and on contextual information to maintain coherence. Our model incorporates these elements through the proposed deformable transformer-based architecture. Temporal information is leveraged via multi-scale temporal convolutional layers that enable the extraction of features at varying time resolutions, capturing the precise start and end of events. Contextual relationships are modeled using the deformable attention mechanism, which dynamically attends to relevant temporal regions, allowing the system to refine predictions based on localized action patterns. These components work in tandem to disambiguate concurrent events and generate distinct, temporally aligned captions, even in complex multi-event scenarios. Moreover, our model integrates the multi-scale

temporal convolutional layers to aggregate features across consecutive frames, enabling the detection of temporal patterns that extend beyond individual frame boundaries.

Our deformable transformer, as described in [32], is an encoder-decoder framework that utilizes multi-scale deformable attention (MSDeformAttn). For our set of multi-scale feature maps  $\{\mathbf{x}^l\}_{l=1}^L$  where  $\mathbf{x}^l \in \mathbb{R}^{C \times H_l \times W_l}$ , a query element  $\mathbf{q}_j$ , and a normalized reference point  $\mathbf{p}_q \in [0, 1]^2$ , MSDeformAttn produces a context vector via a weighted sum of  $K \times L$  sampling points across the feature maps at  $L$  scales:

$$\text{MSDeformAttn}(\mathbf{q}_j, \mathbf{p}_j, \{\mathbf{x}^l\}_{l=1}^L) = \sum_{l=1}^L \sum_{k=1}^K A_{jkl} \mathbf{W} \mathbf{x}_{\tilde{\mathbf{p}}_{jkl}}^l, \\ \tilde{\mathbf{p}}_{jkl} = \phi_l(\mathbf{p}_j) + \Delta \mathbf{p}_{jkl}.$$

In this equation,  $\tilde{\mathbf{p}}_{jkl}$  and  $A_{jkl}$  denote the position and attention weight of the  $k$ -th sampled key at the  $l$ -th scale for the  $j$ -th query element, respectively.  $\mathbf{W}$  represents the projection matrix for key elements, and  $\phi_l$  maps the normalized reference points into the feature map at the  $l$ -th level. The sampling offsets  $\Delta \mathbf{p}_{jkl}$  are relative to  $\phi_l(\mathbf{p}_j)$ . Both  $A_{jkl}$  and  $\Delta \mathbf{p}_{jkl}$  are determined through linear projection onto the query element. Overall, in our deformable transformer, self-attention modules in the transformer encoder and cross-attention modules in the transformer decoder are replaced with deformable attention modules.

The decoding network comprises a deformable transformer decoder and three parallel components, leveraging the strategy in [10]: a captioning head for generating captions, a localization head for predicting event boundaries with confidence scores, and an event counter for estimating the number of events. The decoder’s objective is to directly query event-level features from the frame features using  $N$  learnable embeddings (referred to as event queries)  $\{\mathbf{q}_j\}_{j=1}^N$  and their associated scalar reference points  $p_j$ . The reference point  $p_j$  is obtained through a linear projection followed by a sigmoid activation applied to  $\mathbf{q}_j$ . These event queries and reference points act as initial estimates for the events’ features and locations (center points) and are iteratively refined at each decoding layer. The refined query features and reference points are denoted as  $\tilde{\mathbf{q}}_j$  and  $\tilde{\mathbf{p}}_j$ , respectively.

Our **localization head** produces a box prediction and a binary classification for each event query. The box prediction task aims to determine the 2D relative offsets (center and length) of the ground-truth segment with respect to the reference point. Binary classification generates the foreground confidence for each event query. Both the box prediction and binary classification are carried out using multi-layer perceptrons. This process results in a set of tuples  $[t_{j1}, t_{jf}, \alpha_j^{\text{loc}}]_{j=1}^N$  that represent the detected events, with  $t_{j1}$  and  $t_{jf}$  the initial and final times, and where  $\alpha_j^{\text{loc}}$  is the localization confidence of the event query  $\tilde{\mathbf{q}}_j$ .

Instead of using a two-stage scheme, our LVC employs enhanced event query representations in parallel localization and captioning heads, allowing these two subtasks to be closely related. LVC directly produces a set of events with an appropriate size without relying on heuristic techniques to eliminate redundancy. Within our deformable transformer (see Figure 2), *Event Queries* are produced and introduced into the

decoder. Each of these queries will result in a prediction for a caption. For all experiments, we use a total of 10 queries, although in the ablation studies we evaluate the impact of this parameter.

Our **captioning head** feeds  $\tilde{\mathbf{q}}_j$  into a vanilla LSTM at each timestamp. The word  $w_{jt}$  is predicted by a fully connected layer followed by a softmax activation over the hidden state  $\mathbf{h}_{jt}$  of the LSTM. Note that this LSTM operates only within the video segment of duration  $\Delta t$  that our LVC model is analyzing.

The **event counter head** predicts the number of events using a max-pooling layer and a fully connected layer with softmax activation, producing a fixed-size vector  $\mathbf{r}_{\text{len}}$ , where each value refers to the possibility of a specific number. The predicted event number is obtained by  $\text{argmax}(\mathbf{r}_{\text{len}})$ . Top  $N_{\text{set}}$  events are selected based on accurate boundaries and captions.

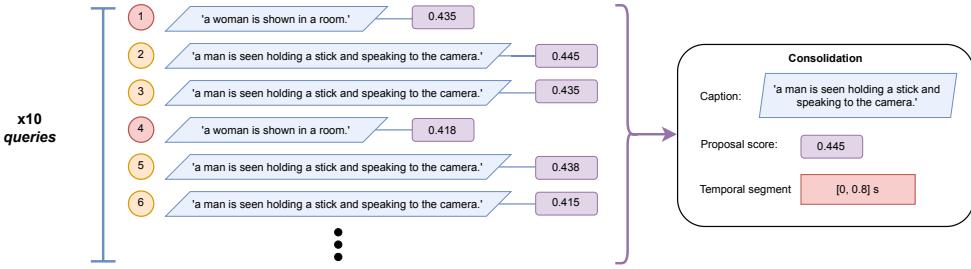
Confidence for each event query is calculated by combining two components: location confidence and a modulated caption confidence. The location confidence evaluates the spatial accuracy of the event, while the modulated caption confidence incorporates adjustments for variability in sentence length. During training, our LVC model generates a set of  $N$  predicted events, each comprising a spatial location and an associated caption. To associate these predictions with the ground truth, the Hungarian algorithm is employed, solving a bipartite matching problem that optimizes a comprehensive cost function. This cost function integrates two primary components: the Generalized Intersection over Union (GIoU) loss [35] for spatial alignment and a focal loss [36] for classification accuracy.

Our total training loss,  $\mathcal{L}_{\text{total}}$ , is computed as a weighted combination of several loss terms:  $\mathcal{L}_{\text{total}} = \lambda_{\text{GIoU}} \mathcal{L}_{\text{GIoU}} + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}} + \lambda_{\text{count}} \mathcal{L}_{\text{count}} + \lambda_{\text{cap}} \mathcal{L}_{\text{cap}}$ , where  $\mathcal{L}_{\text{GIoU}}$ ,  $\mathcal{L}_{\text{cls}}$ ,  $\mathcal{L}_{\text{count}}$ , and  $\mathcal{L}_{\text{cap}}$  represent the GIoU loss, classification loss (i.e. focal loss), counting loss, and caption loss, respectively. Our counting loss is the cross-entropy loss between the estimated count distribution and the one in the ground-truth. For the captioning loss we use the cross-entropy computed between the word probability estimated by our model and the ground-truth normalized by the length of the caption. The hyper-parameters  $\lambda_{\text{GIoU}}$ ,  $\lambda_{\text{cls}}$ ,  $\lambda_{\text{count}}$ , and  $\lambda_{\text{cap}}$  control the relative contributions of each loss term. For our experiments we use  $\lambda_{\text{GIoU}} = 2$ , and  $\lambda_{\text{cls}} = \lambda_{\text{count}} = \lambda_{\text{cap}} = 1$ .

While offline models for dense video captioning can access all available information by analyzing the entire video before generating the captions, our LVC model only has access to the information available in the video segment of duration  $\Delta t$ . This limitation can hinder the quality of the subtitles because a short video segment might not provide enough context. In our LVC model, captions are generated in an online manner based exclusively on the video data available up to the current time step. Once a caption is produced for a given video segment, it remains immutable and is not revised as new frames are processed. This design choice enforces the causality constraints essential to the LVC task, ensuring that each caption accurately reflects the observed content at the moment of generation without relying on future information. This approach distinctly differentiates LVC from traditional offline methods, which have access to the entire video sequence and can even refine captions retrospectively.

Therefore, to improve the quality of the captions, we have included a **temporal filtering for the caption consolidation**, as shown in Figure 3. First, our deformable

transformer uses a query mechanism comprising  $N$  queries per video segment of duration  $\Delta t$ . Subsequently, our filtering module accepts the predictions generated by these event queries,  $\{\mathbf{q}_j\}_{j=1}^N$ , as input. Each query produces a predicted caption  $c_j$ , an associated temporal segment  $[t_{j1}, t_{jf}]$ , and a confidence score  $\alpha_j$ , resulting in a set of tuples  $\{(t_{j1}, t_{jf}, c_j, \alpha_j)\}_{j=1}^N$ . Considering the temporal overlap between the predictions, the consolidation module orchestrates a voting mechanism to determine the final captions for the given video segment. Specifically, the temporal filtering we propose consists in a module that evaluates the frequency of occurrence of each overlapped predicted caption among the set of predictions. Let  $f(c)$  represent this frequency for a particular caption  $c$ , and  $w(c)$  its maximum score. The ultimate caption  $c_*$  for the set of temporally overlapped predictions is selected as the one with the highest frequency:  $c_* = \arg \max_{c \in \{c_1, \dots, c_N\}} f(c)$ . Being the final tuple for the prediction as follows:  $\{(t_{*1}, t_{*f}, c_*, w(c_*))\}$ .



**Fig. 3** Example of caption consolidation for a video segment.

### 3.3 Novel Evaluation Metric for Live Video Captioning: the Live Score

Dense video captioning models, by operating in offline mode, have traditionally been evaluated using offline metrics. The new LVC paradigm we propose necessitates the development of a new evaluation metric with an online nature. We begin by describing the typical offline evaluation scheme and then highlight its main limitations for LVC systems, thereby introducing the properties that an online evaluation metric must possess. Next, we introduce the formulation associated with the new proposed metric, the Live Score, including all of its variants.

#### 3.3.1 Online evaluation metric properties

The main characteristics of traditional offline dense video captioning metrics are as follows. All the information from the video annotations is introduced at once in the metrics. Then, the scorer rates the entire input video regardless of its duration. This scenario is quite different from that of LVC, where dense caption predictions with their associated timestamps arrive in temporal blocks, whose duration is less than that of the full video being processed. Moreover, the offline metrics cast a score that corresponds

to the average score for a full video. This fact does not allow us to *observe* the temporal evolution in the performance of the models, and this property is interesting for LVC. In other words, we need an online nature metric that evolves with the video and reflects the online accuracy of LVC systems.

Therefore, the characteristics that an online metric for LVC must fulfill are as follows:

- **Video stream-based scoring:** Ability to obtain scores from a video stream, therefore no access to a whole video is needed.
- **Causality compliance:** The metric should not have access to future information, only to what is being processed at the moment and what has already been processed.
- **Memory-aware:** Implementation of a record of the scores obtained in the video. This will be used to calculate subsequent results and allow for an evolution over time.

### 3.3.2 The Live Score

For the new paradigm of LVC, we propose a purely online metric: the Live Score (LS). In short, it is an adaptation of the various scores used in offline metrics, but tailored to process video streams online, causally, and considering the history of the video, as we have specified in the previous section. Evaluating captioning quality in live video scenarios requires continuous assessment, as the accuracy of captions must be ensured at each stage of video playback. We propose an experimental evaluation guided this novel LS metric, which tracks the temporal evolution of caption accuracy as the video progresses. The LS metric allows us to evaluate caption performance at each moment, taking into account both the immediate accuracy and the cumulative quality of predictions. This continuous evaluation is crucial in live scenarios, where the model must maintain high performance throughout the video's duration, ensuring that captions remain relevant and precise as events unfold.

We begin with the necessary mathematical formulation to define the LS. Let an LVC model aim to produce a series of caption predictions by analyzing an input video stream every  $\Delta t$  seconds. Note that  $\Delta t$  will be the only configurable parameter of the new metric LS. We define  $C_i$  as the set of captions generated by the model LVC when presented with a video  $V_i$ :

$$LVC(\Delta t, V_i) \Rightarrow C_i = \{[t_{1i}, t_{1f}, c_1, \alpha_1], [t_{2i}, t_{2f}, c_2, \alpha_2], \dots, [t_{ni}, t_{nf}, c_n, \alpha_n]\}, \quad (1)$$

where  $t_{ni}$  and  $t_{nf}$  are the start and end times of each timestamp, respectively,  $c_n$  contains the predictions for the captions, and  $\alpha_n$  encodes the confidence assigned by the LVC model to each caption.

The LS metric will process the data in  $C_i$  online, providing a score  $\gamma_{t'}$  for each timestamp  $t'$ , with a resolution of  $\Delta t$  seconds. We propose to combine our LS metric with any of the traditional scorers for video captioning (see Figure 4). This scorer is now evaluated continuously, and our LS metric allows for the observation of its evolution, instantaneously. The scorers are responsible for comparing the similarity

between the predicted and annotated captions. The ones we have integrated into our LS metric and that have been used in experiments are as follows:

- METEOR [37]: It is an automatic translation quality evaluation metric that calculates word, phrase, and synonym similarity scores.
- Bleu4 [38]: Automatic evaluation metric used in text generation and machine translation. It emphasizes the accuracy of matching four-word *n-grams* and evaluates the similarity between the generated output and human references using the count of matching *n-grams*.
- Rouge-L [39]: Automatic evaluation metric primarily used in automatic text summarization. The generated summary and the reference summary are compared using word count and summary length.

To reflect a continuous temporal evolution, when we have the score calculated at  $t'$ , we compute the mean with all previous scores, so that the LS metric is formulated as follows:

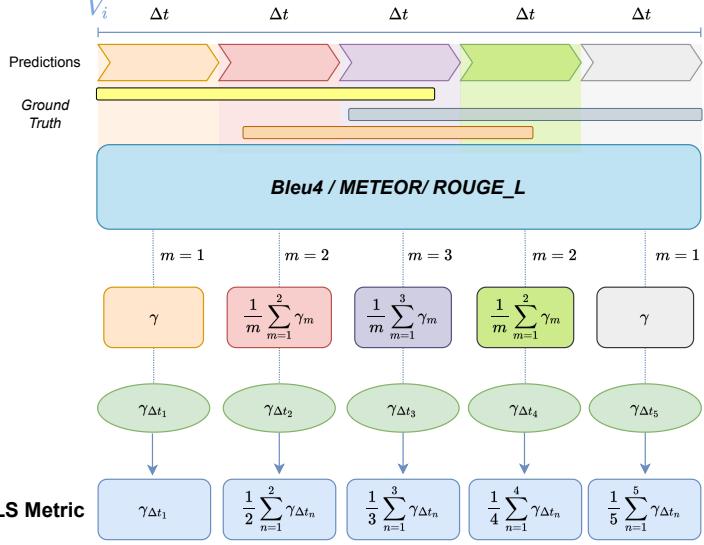
$$\text{LS}(t', \text{LVC}(\Delta t, V_i)) = \frac{\sum_{n=1}^K \gamma_{t'_n}}{K}, \quad (2)$$

where  $t'_n = n \cdot \Delta t$ , the numerator corresponds to the sum of all scores calculated up to the current moment  $t'$ , and  $K = \frac{t'}{\Delta t}$ .

As it is shown in Figure 4, we can have multiple ground-truth captions associated with the video segment we are processing. Remember we are dealing with the *dense* video captioning problem, hence this situation is possible. Our metric will produce a score between the predictions and each annotated caption, resulting the final score  $\gamma_{t'}$  as the average of all generated scores for that segment. We show in Figure 4 a graphical example, where the LS metric is used to process a video  $V_i$  segmented into fragments with length  $\Delta t$ , each one containing  $m$  associated captions.

The proposed LS metric has an online nature, meeting all the requirements detailed in Section 3.3.1. However, it does not take into account the influence of false positives, i.e., predictions of captions that do not appear in the annotations available in the database. In other words, the proposed metric does not include any calibration mechanism with respect to false positives. In a realistic scenario for the LVC problem, such as generating captions for a live video stream from a surveillance camera, it is highly likely that there are large portions of the video where no action is occurring. Thus, in an LVC model applied to a video surveillance system, we must avoid at all costs the model generating captions for events that have not occurred. Indeed, false-positive captions pose a significant challenge in live video captioning where they can disrupt the user experience by generating frequent, inaccurate descriptions. An LVC system will be accurate if it provides accurate captions, but also if it only provides captions when something relevant is happening in the video.

To calibrate our metric and make it sensitive to false positives, we propose integrating a penalty for false positives into the LS. This new version of the metric is



**Fig. 4** The LS metric. It allows for an online and continuous evaluation of a video stream, analyzed every  $\Delta t$  seconds. Our metric allows for the integration of any scorer (e.g. METEOR, Bleu4 or Rouge-L) in the online or live evaluation.

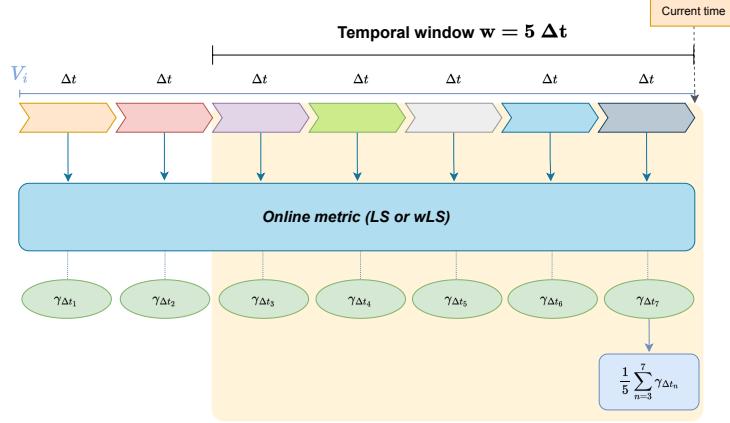
called weighted-LS (wLS), and is formulated as follows:

$$wLS(t', LVC(\Delta t, V_i)) = \frac{\sum_{n=1}^K \gamma_{t'_n}}{K} \cdot e^{-\beta}, \quad (3)$$

$$\beta = \frac{\sum_{n=1}^K fp(t'_n)}{K}, \quad (4)$$

where we have added the correction factor  $\beta$ , dependent on  $fp(t'_n)$ , which is the number false positives corresponding to the video segment associated to  $t'_n$ . By incorporating this weighted metric, our experimental evaluation pipeline enhances the reliability of live captioning systems, ensuring a more seamless and coherent user experience.

The two new proposed metrics, LS and its calibrated version wLS, allow for the online and continuous evaluation of what happens in a video up to time  $t'$ , considering the entire history of the video from  $t = 0$ . It may happen that the evaluation process starts with predictions that have very low scores and gradually improve over time, or vice versa. In such scenarios, since the metric is calculated based on *all* previous scores obtained by the system from  $t = 0$ , it always considers the entire temporal timeline, and the metric may fail to reflect the system's most *recent* behavior. To address this issue, we propose a version for both LS and wLS that considers only a fixed temporal history window for computing the performance of the LVC system at



**Fig. 5** Operation of the online metric with fixed temporal window history. We observe how that temporal window moves along the video timeline. The window, with size  $w = 5$ , encompasses the scores that will be considered to compute the score associated with the current instant. The first two slots have been discarded. The diagram has been simplified for ease of understanding, but the calculation of scores for each  $\Delta t$  is the same as in the previous scenarios.

time  $t'$ , covering only the interval  $[t' - w\Delta t, t']$ . Here,  $w$  defines the size of this fixed temporal window used to compute the metrics LS and wLS. By updating the metric considering only the fixed temporal history window, we allow it to evolve, reflecting the current performance of the LVC model. In Figure 5, we illustrate the calculation process for a temporal window size of  $w = 5$ .

Incorporating the fixed temporal history window, the formulations of the previously described metrics are as follows:

$$hLS(t', LVC(\Delta t, V_i)) = \frac{\sum_{n=\max(1, K+1-w)}^K \gamma_{t'_n}}{K}, \quad (5)$$

$$hwLS(t', LVC(\Delta t, V_i)) = \frac{\sum_{n=\max(1, K+1-w)}^K \gamma_{t'_n}}{K} \cdot e^{-\beta}, \quad (6)$$

$$\beta = \frac{\sum_{n=\max(1, K+1-w)}^K fp(t'_n)}{K}. \quad (7)$$

## 4 Experimental Evaluation

In this section, we provide details of the experimental evaluation designed for the proposed novel LVC problem.

We start in Section 4.1 with a description of the experimental setup, where we outline the database used and the adaptations made to it for the evaluation of the live models. Subsequently, in Sections 4.2-4.6, we include both qualitative and quantitative results of all our experiments. The questions we want to address with the proposed experimental evaluation are as follows:

1. Are offline experimental evaluation environments adequate for LVC models?
2. Are the proposed new metrics suitable for the LVC problem, and do they allow to judge the temporal evolution of LVC systems?
3. What is the performance of the proposed model for LVC? With respect to the state of the art in offline models, how does the proposed LVC approach perform?

#### 4.1 Experimental setup

We have used for our experimental evaluation the *ActivityNet Captions* [13] dataset. This database is actually a subset of data from *ActivityNet* [40]. Specifically, *ActivityNet Captions* consists of a set of 20,000 videos totaling 849 hours of video with a total of 100,000 descriptions, each with its start and end timestamp. On average, each annotated video contains 3.65 localized phrases, with each phrase averaging 13.48 words. The videos were generated at a rate of 30 frames per second. For our experiments we have chosen the validation set, which contains 4,926 videos.

The different  $\Delta t$  values used in the experiments were (24, 48, 72, 96, 120, 150) in frame numbers. We justify this choice of values for  $\Delta t$  because all of them represent a reasonable temporal length for online applications. If the values were greater, the delay between predictions would be too high, making them unsuitable for consideration as *live* models.

#### 4.2 Evaluating LVC models with off-line metrics

Are *offline* experimental evaluation metrics suitable for the novel LVC problem? This is the question we want to specifically address in this section. We set up an experimental evaluation scenario in which we use the official *offline* metrics provided in the ActivityNet Challenge 2018 [41], on the online predictions generated by our LVC system. For localization performance, the average precision and average recall across intersection over union at different thresholds are used. For dense captioning performance, the official evaluation tool provided by ActivityNet Challenge 2018 is followed, which calculates the average precision measured by BLEU4, METEOR, and ROUGE\_L scorers, of the matched pairs between generated captions and the ground truth across intersection over union thresholds of 0.3, 0.5, 0.7 and 0.9.

In Table 2, we first present the results in terms of dense captioning accuracy. We provide a detailed comparison between the results of our online LVC model and those offered by the offline models PDVC [10] and Vid2Seq [9], that are the state of the art for the dense video captioning problem.

Analyzing these results, we can draw the following conclusions. The first one is that offline metrics favor offline models. Our live system achieves low performance in some metrics. The reason is clear: these metrics filter the captions generated by the models based on intersection-over-union thresholds, as we have seen, and the predictions of

Model	Live	$\Delta t$	Features	Bleu4	METEOR	ROUGE_L
PDVC [10]	$\times$		C3D	1.65	7.50	-
			TSN	1.78	7.96	-
			TSP	<b>2.17</b>	<b>9.03</b>	-
PDVC_light [10]	$\times$		C3D	1.51	7.11	-
			TSN	1.66	7.97	-
			TSP	<b>1.77</b>	<b>8.55</b>	-
Vid2Seq [9]	$\times$		CLIP	-	8.5	-
LVC (Ours)	$\checkmark$	72	TSP	0.13	0.14	0.15
				0.47	0.45	0.51
				0.85	0.75	0.91
				1.21	1.03	1.29
				1.55	1.27	1.63
				<b>2.01</b>	<b>1.56</b>	<b>2.08</b>

**Table 2** Experimental comparison in terms of accuracy of the captions generated by offline models and our online model LVC using traditional offline metrics for the *ActivityNet Captions* dataset, using the validation set of videos.

our LVC model are too short in temporal duration ( $\Delta t$  is the value), so many of them do not survive this filtering and are naturally discarded by the offline metrics. The second conclusion, related to the first one, is that offline metrics tend to improve as we increase the parameter  $\Delta t$  in our LVC model. The metrics were designed to work in offline scenarios where models can and should see the entire video first, and then generate all caption predictions. This favors the generation of captions of much longer duration than those that can be generated by our LVC model (with a  $\Delta t$  limit). In fact, the offline dense captions can even occupy large temporal portions of the video.

We can also compare offline and live models in terms of the accuracy of temporal localization of the captions, again using traditional offline metrics: precision and recall. In Table 3, we present this detailed analysis, and we can observe that the results of event localization obtained for our predictions using these offline metrics are not satisfactory. Again, the metrics improve as  $\Delta t$  increases in our LVC model. The explanation is similar to what we have provided for the previous metrics: our caption predictions are associated with video segments of duration  $\Delta t$ , which causes them not to meet the intersection-over-union criteria employed by the offline metrics. Observe the low performance when the threshold of 0.9 is used for the intersection-over-union.

### 4.3 Analysis using modified offline annotations

One might argue that offline metrics could still be applicable in an online scenario if the video annotations are adjusted to align with the duration used by LVC models. We have also performed this analysis for completeness, although we anticipate that this approach has significant drawbacks.

As an alternative to designing an online metric, one can attempt to use offline metrics but on a dataset where annotations have been modified to achieve an *online*

Model	Live	Recall					Precision					
		0.3	0.5	0.7	0.9	avg	0.3	0.5	0.7	0.9	avg	
MFT [42]	✗	46.18	29.76	15.54	5.77	24.31	86.34	68.79	38.30	12.19	51.41	
SDVC [43]	✗	<b>93.41</b>	<b>76.40</b>	42.40	10.10	55.58	96.71	77.73	<b>44.84</b>	10.99	57.57	
PDVC.light [10]	✗	88.78	71.74	<b>45.70</b>	<b>17.45</b>	<b>55.92</b>	96.83	78.01	41.05	<b>14.69</b>	57.65	
PDVC [10]	✗	89.47	71.91	44.63	15.67	55.42	<b>97.16</b>	<b>78.09</b>	42.68	14.40	<b>58.07</b>	
Vid2Seq [9]	✗	-	-	-	-	52.7	-	-	-	-	53.9	
<b>Δt</b>												
LVC - (ours)	✓	24	7.60	2.72	0.89	0.16	2.84	1.81	0.45	0.21	0.06	0.63
		48	17.57	7.61	2.55	0.47	7.05	5.88	1.98	0.78	0.15	2.20
		72	25.40	12.30	4.61	0.92	10.81	10.14	3.91	1.79	0.42	4.07
		96	31.76	16.45	6.42	1.31	13.98	14.21	6.10	2.74	0.63	5.92
		120	36.70	19.34	7.67	1.74	16.36	18.17	7.94	3.58	0.82	7.63
		150	<b>42.50</b>	<b>22.87</b>	<b>9.67</b>	<b>2.25</b>	<b>19.32</b>	<b>23.11</b>	<b>10.24</b>	<b>4.77</b>	<b>1.23</b>	<b>9.84</b>

**Table 3** Caption localization for the validation video set of *ActivityNet Captions*, using traditional offline metrics. Comparison with the state of the art offline models.

	Model	Δt	Recall					Precision				
			0.3	0.5	0.7	0.9	avg	0.3	0.5	0.7	0.9	avg
Original Annotations	LVC	24	7.60	2.72	0.89	0.16	2.84	1.81	0.45	0.21	0.06	0.63
		48	17.57	7.61	2.55	0.47	7.05	5.88	1.98	0.78	0.15	2.20
		72	25.40	12.30	4.61	0.92	10.81	10.14	3.91	1.79	0.42	4.07
		96	31.76	16.45	6.42	1.31	13.98	14.21	6.10	2.74	0.63	5.92
		120	36.70	19.34	7.67	1.74	16.36	18.17	7.94	3.58	0.82	7.63
		150	42.50	22.87	9.67	2.25	<b>19.32</b>	23.11	10.24	4.77	1.23	<b>9.84</b>
Modified Annotations	LVC	24	90.97	90.08	89.24	88.32	<b>90.16</b>	97.63	97.32	96.85	96.23	<b>97.18</b>
		48	90.36	88.73	87.10	85.14	88.75	97.79	97.33	96.24	94.66	96.82
		72	89.85	87.46	85.05	82.30	87.46	97.88	97.22	95.46	93.18	96.39
		96	89.33	86.25	83.28	79.74	86.29	97.96	97.15	94.86	91.74	96.00
		120	88.88	85.02	81.48	77.43	85.16	97.99	97.01	94.06	90.42	95.57
		150	88.28	83.74	79.58	74.72	83.89	97.98	96.84	93.42	88.85	95.11

**Table 4** Comparison of results using original annotations or annotations adapted to live models, in terms of caption localization.

*appearance*. In other words, it involves taking the temporal annotations for each caption and dividing them into small temporal segments that match the temporal window used by live video captioning models, i.e.,  $\Delta t$ .

We have automated a process to modify the annotations provided in the validation set of the *ActivityNet Captions* database. Once the modified annotations are generated, traditional metrics for offline video captioning are employed, and the results are as follows. In Table 4, we present the results in terms of caption localization in this new scenario and compare it with the performance obtained with the original annotations. Note that by splitting the provided annotations, we ensure that the predictions of our live model are not filtered out because they do not meet the intersection over union criterion.

It is interesting to observe how adapting the annotations to the live solutions results in a considerable improvement in localization metrics. The best average precision jumps from 9.84% to 97.18%, while the best average recall reaches 90.16% from only 19.32%.

We also report the performance of our model for the three different metrics used to evaluate the precision of the captions, see Table 5. Again, we compare the performance

Scenario	$\Delta t$	Bleu4	METEOR	ROUGE_L
Original Annotations	24	0.13	0.14	0.15
	48	0.47	0.45	0.51
	72	0.85	0.75	0.91
	96	1.21	1.03	1.29
	120	1.55	1.27	1.63
	150	<b>2.01</b>	<b>1.56</b>	<b>2.08</b>
Modified Annotations	24	<b>18.18</b>	<b>8.71</b>	<b>18.27</b>
	48	17.79	8.51	17.89
	72	17.34	8.28	17.51
	96	17.05	8.12	17.23
	120	16.86	8.02	17.05
	150	16.50	7.83	16.74

**Table 5** Comparison between results using original annotations and annotations adapted to the live scenario for *ActivityNet Captions*, employing LVC as the model.

when the original annotations and the adapted ones are used. We can observe that in this experiment the trend is that, the lower  $\Delta t$  is, the higher the results are.

In conclusion, adapting annotations to how live models work has a positive impact on their evaluation. However, the proposed adaptation has the drawback that the annotation must be dynamically adjusted to the temporal window  $\Delta t$  being used, to then employ traditional offline metrics. Furthermore, it is not an evaluation strategy that naturally provides a metric that allows us to observe the temporal evolution of the model. In other words, these offline metrics do not evolve over time with the video, a fundamental aspect for the novel LVC problem. All these drawbacks are clearly addressed by the new metrics proposed in this work.

We conclude that the analysis performed in this section is crucial to justify the need for a new evaluation metric for live models, so that we can evaluate them efficiently and fairly, as we shown in the following section.

#### 4.4 Evaluating LVC models with the novel Live Score

One of the main motivations of our work has been to design a new online evaluation metric for live video captioning models, i.e., the LiveScore (LS) (see Section 3.3.2). In this section, we detail all the experimental evaluations carried out using it. Note that we provide a detailed experimental analysis considering the four alternatives proposed for the LS metric:

- Normal operation mode (*Live Score* - LS).
- Mode with a correction factor (*Weighted Live Score* - wLS).
- Mode with history in a memory window (*LS with History Window* - hLS).
- Combined mode with correction factor and history in a memory window (*Weighted LS with History Window* - whLS).

In Table 6, we show the mean obtained by the novel LS when integrated with the different scorers Bleu4, METEOR, and ROUGE\_L. Based on the results obtained, we can draw the following conclusions. First, it is observed that the metrics increase as

Scorer	LVC - $\Delta t$					
	24	48	72	96	120	150
LS	Bleu4	18.13	19.00	19.29	19.38	<b>19.45</b>
	METEOR	8.98	9.32	9.37	9.38	<b>9.40</b>
	ROUGE_L	19.84	20.81	21.14	21.21	<b>21.29</b>
wLS	Bleu4	17.12	18.19	18.61	18.80	<b>18.96</b>
	METEOR	8.45	8.91	9.03	9.09	<b>9.16</b>
	ROUGE_L	18.71	19.90	20.36	20.55	<b>20.74</b>

**Table 6** Evaluation for the LVC model using the new online metric.

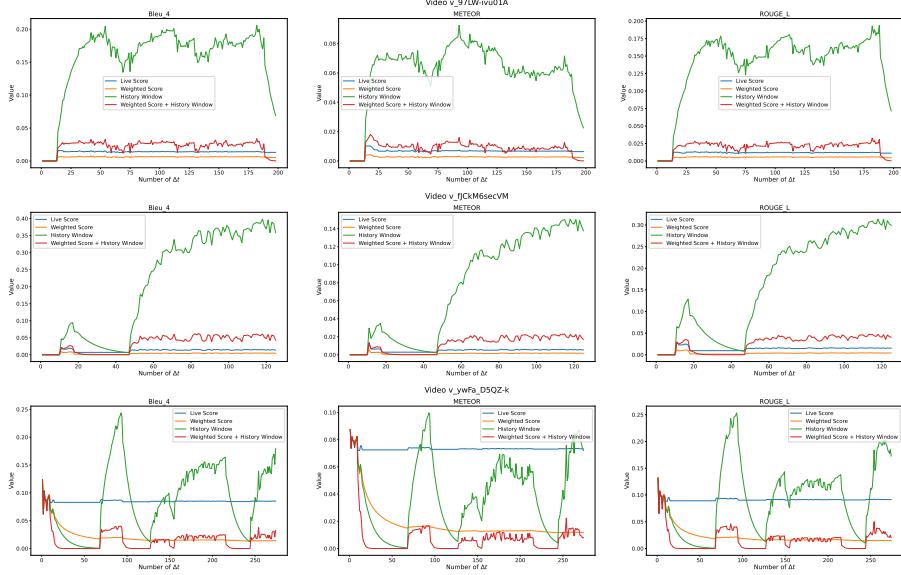
LS: Live Score. wLS: Weighted LS. We evaluate the LVC model for different values of the parameter  $\Delta t$ , as well as integrating different scorers (Bleu4, METEOR, and ROUGE\_L).

the parameter  $\Delta t$  increases. This makes sense, as the larger the temporal window, the LVC model is able to see more portions of the video and offer a better description in the caption. However, values higher than  $\Delta t = 120$  do not seem to offer a significant improvement. Second, the results reported by the online metric are considerably higher than those obtained by the offline metrics. This becomes clear when comparing the results between Table 2 and Table 6, where the results with the offline and online metrics are shown, respectively. For example, for  $\Delta t = 120$ , the Bleu4 scorer goes from 1.55 to 19.45 for the LS version, or to 18.96 for the wLS version. The increase experienced for the rest of the scorers is somewhat similar. This demonstrates the suitability of the new metric for the online scenario. Third, when comparing between the LS and wLS versions, we observe how LS offers higher results than the wLS version. This is because false positives that are generated are not considered by LS, but only by the wLS version, thus offering lower scores, but more adjusted to the actual performance of the LVC model. Fourth, given the results offered by the different scorers and the LS and wLS metrics, the LVC model achieves the best results for  $\Delta t = 120$ , with consensus among all scorers.

Finally, we include the analysis of the temporal evolution of the performance of our LVC model using all our novel evaluation metrics. This analysis demonstrates that the new metrics enable us to generate graphs that track the temporal evolution of different scores. Such tracking is crucial for LVC models and cannot be achieved with traditional offline metrics. Note that we include the hLS and whLS variations of the metric in this analysis, which are specifically designed to continuously visualize the accuracy of LVC systems.

In Figures 6 to 11, we show the results obtained for 3 videos and all the values of the parameter  $\Delta t$  used. Analyzing these graphs, we can draw the following conclusions. Firstly, the granularity of the metrics naturally increases as we decrease the parameter  $\Delta t$ . This allows us to control the speed at which captions are generated and the speed at which their accuracy is evaluated. Secondly, we have also included LS and wLS metrics without the temporal window option in these graphs. We can observe how for all videos and all scorers, the wLS version is always more conservative, reporting lower or equal scores, as it applies a penalty based on false positives. The same behavior is observed when comparing hLS and whLS, with the latter offering the most conservative

scores. It is also noticeable that all versions start by reporting exactly the same values, beginning to diverge when the history window comes into play. As the third conclusion, perhaps the most important one, these plots show how the inclusion of a history window in our metric allows the model to exhibit its recovery in terms of caption generation accuracy. It no longer considers the entire past, only the recent local past, so that the scorers can increase or recover as the LVC model chains more and more correct caption predictions. In fact, observing a downward trend in the hLS and whLS metrics is an important indicator of how poorly the caption generator system is performing.

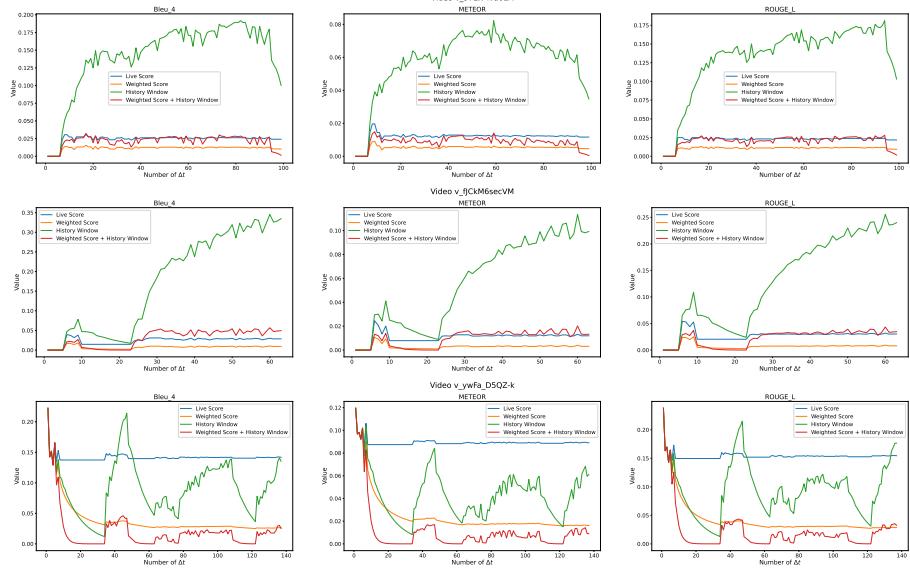


**Fig. 6** Temporal evolution of the designed online metrics for the LVC model with  $\Delta t = 24$ . Results are shown for 3 videos from ActivityNet Captions.

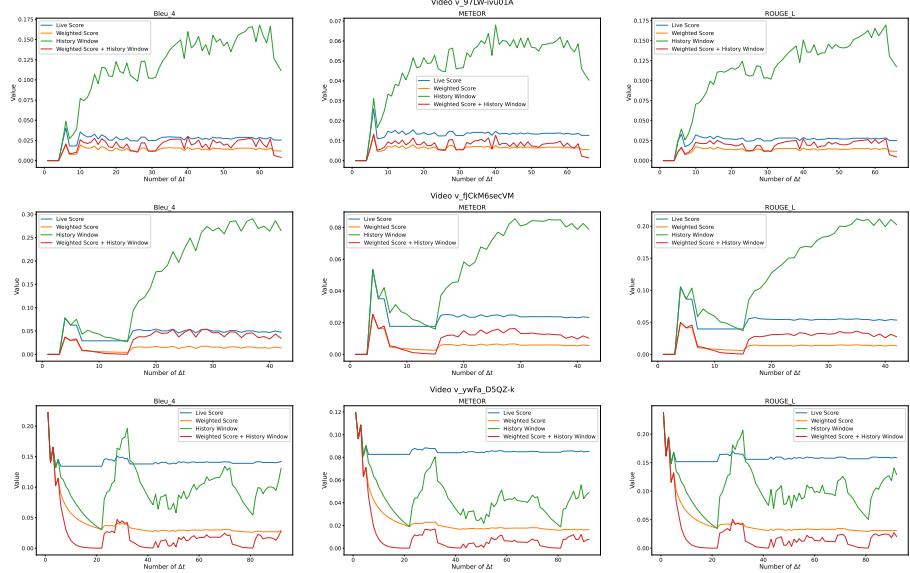
#### 4.4.1 Discussion

While novel LS metric provides a robust online evaluation framework for LVC systems, it is not used directly during their training, as it can be observed in previous experiments. LS calculates a cumulative score over time, capturing the real-time and historical accuracy of predictions, which contrasts with the frame-by-frame optimization process used during training of our LVC model. Training objectives typically rely on instantaneous feedback, that are easier to optimize with stochastic gradient descent. Exploring methods to incorporate LS principles into a training objective remains an intriguing avenue for future research, potentially bridging the gap between evaluation and model optimization.

Another interesting aspect to analyze in the context of LVC is the quality of the annotations in the training datasets. The robustness of video captioning models heavily

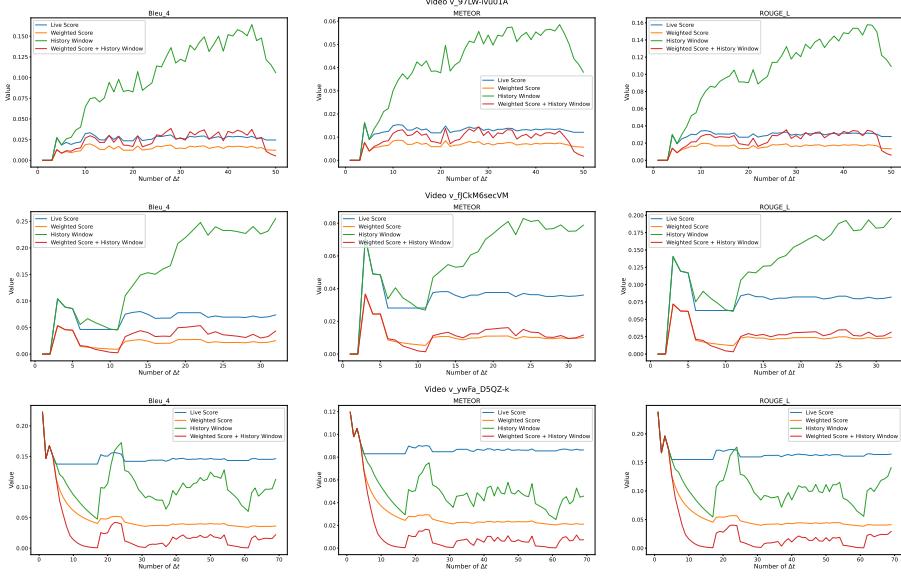


**Fig. 7** Temporal evolution of the designed online metrics for the LVC model with  $\Delta t = 48$ . Results are shown for 3 videos from ActivityNet Captions.

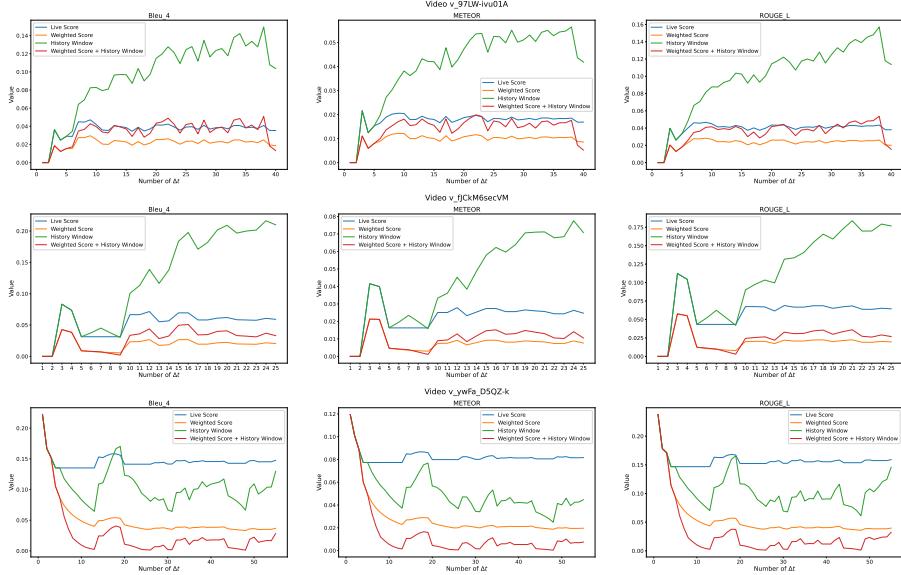


**Fig. 8** Temporal evolution of the designed online metrics for the LVC model with  $\Delta t = 72$ . Results are shown for 6 videos from ActivityNet Captions.

depends on the quality and diversity of annotations in the training datasets. High-quality annotations ensure accurate temporal boundaries and rich semantic details,

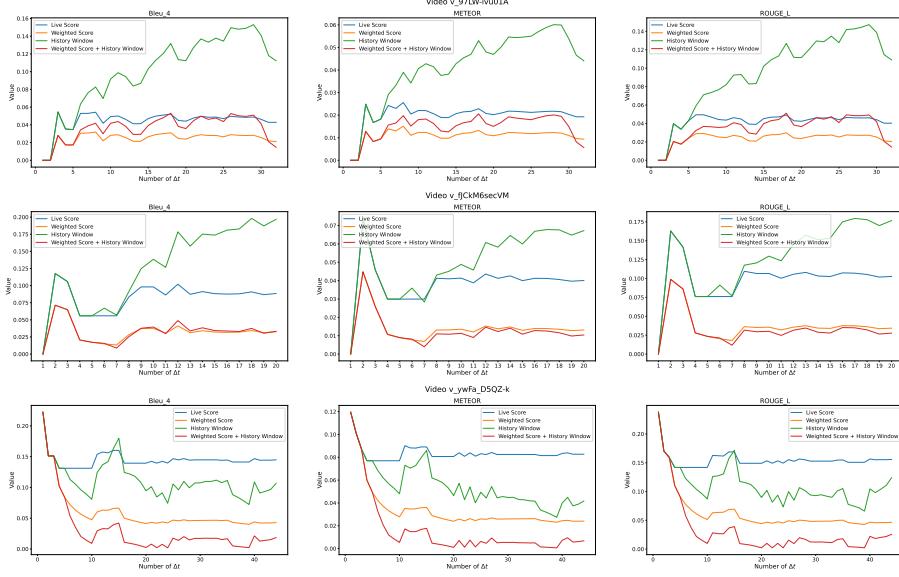


**Fig. 9** Temporal evolution of the designed online metrics for the LVC model with  $\Delta t = 96$ . Results are shown for 3 videos from ActivityNet Captions.



**Fig. 10** Temporal evolution of the designed online metrics for the LVC model with  $\Delta t = 120$ . Results are shown for 3 videos from ActivityNet Captions.

enabling the model to learn fine-grained associations between actions and descriptions. Conversely, noisy or inconsistent annotations can introduce ambiguity, reducing the



**Fig. 11** Temporal evolution of the designed online metrics for the LVC model with  $\Delta t = 150$ . Results are shown for 3 videos from ActivityNet Captions.

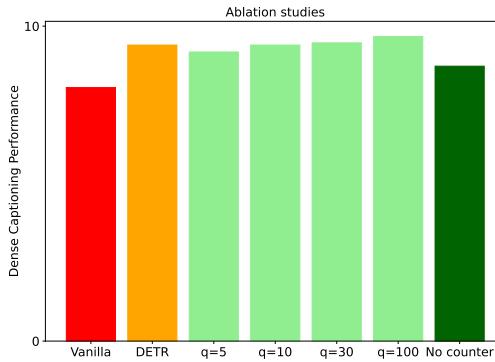
model's ability to capture temporal precision and action differentiation. Similarly, a lack of variety in annotated events or scenarios limits the model's capacity to generalize to unseen contexts, particularly in real-world applications with diverse and dynamic content. These factors are especially critical in LVC, where models must operate with partial observations and strict timing constraints. Addressing these limitations requires the development of datasets with comprehensive, high-quality annotations tailored to the online scenario.

#### 4.4.2 Ablation Studies

In this section, we show the impact of the parameters of our LVC model on the quality of the dense captions. Specifically, we analyze the influence of the following components of our model: the use of a deformable transformer vs. a standard transformer; the number of queries; and the event counter head. For these studies we have used the METEOR metric in conjunction with our LS. Figure 12 shows the results obtained. As can be observed, the use of a deformable transformer significantly improves the results. Regarding the number of queries used in our transformer, the impact is not as substantial, with  $q = 10$  being a fairly efficient compromise. Lastly, it is notable that not using the event counter head leads to a significant drop in the model's performance.

In summary, regarding the architecture used in the proposed LVC model, the use of a deformable transformer is confirmed as the option that provides the best results. The influence of the deformable transformer's specific parameters, such as the number of queries, does not seem to significantly affect the quality of the captions. We believe this is because our model, instead of processing the entire video, works with short segments of the video, due to the online nature of the problem posed by LVC. Finally,

the use of the three simultaneous heads—one for captions, another for localization, and another for considering the estimation of the number of events—appears to deliver the best results.



**Fig. 12** We show the impact of the different parameters and components of our LVC model on the quality of the dense captions when using a standard transformer (Vanilla) vs. a DETR, different values for  $q$ , and when the event counter head is not activated. We report METEOR with the LS metric.

## 4.5 Qualitative analysis

In this section, we present some qualitative results of our LVC system. Specifically, we show two examples corresponding to two videos from the validation set: one demonstrating a good result and the other a poor result. We used  $\Delta t = 150$ , the maximum value we experimented with. This makes it easier to identify points in the graph where there is a change in slope according to the proposed metrics. We demonstrate the operation of the metric using the *Live Score* strategy, integrating the *Bleu4* metric. In both examples, we highlight several points on the graph where notable changes in slope occur. A pronounced change in slope indicates a significant improvement or deterioration in the score compared to previous points. These marked points of interest represent where the predictions either align more closely or diverge from the ground truth. Red shading signifies a greater discrepancy between the prediction and the ground truth, while green shading indicates a closer match.

Figure 13 contains the good case. We can observe that it represents a scenario where the predictions closely resemble the ground-truth. The scores obtained by the metric are quite good, in fact. Figure 14 represents a case where the predictions differ significantly from the ground-truth and the scores are low.

## 4.6 Demo

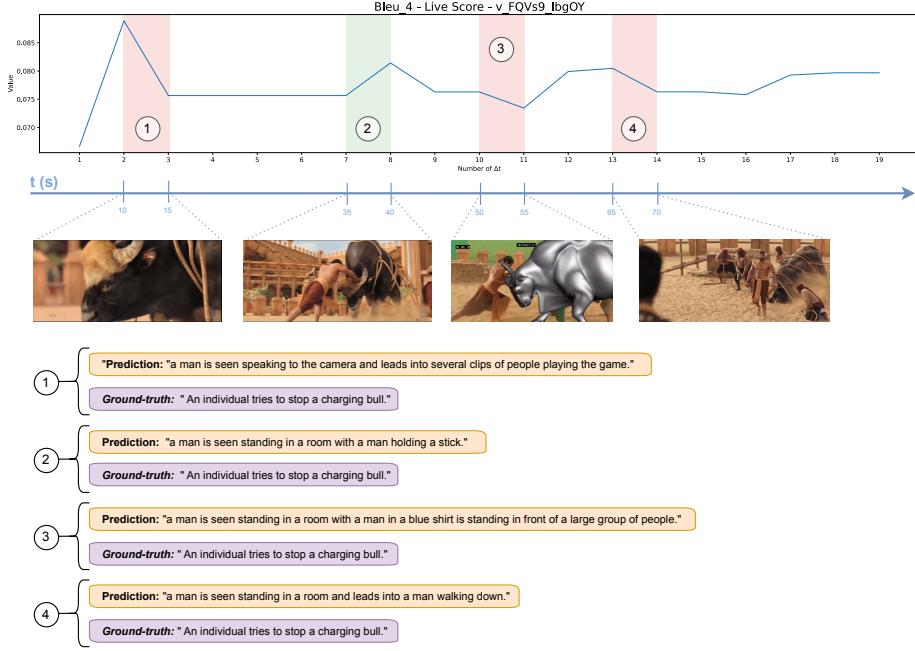
In this section, we showcase the system in operation with a live video stream and highlight its performance. We have implemented a demo that allows for the direct processing of a video stream from a camera using our LVC model.



**Fig. 13 Good Quality Example:** It is observed that the predictions of the LVC system resemble those that are annotated.

To achieve processing speed and provide captions with good quality, we have had to make the following interventions in the LVC model. The demonstrator works by directly accessing a video stream from a webcam. Once launched, the model will begin generating captions on the video stream immediately and continuously, displaying them on the screen. We have implemented a multiprocessing solution with two threads running simultaneously. The multiprocessing implementation is essential to ensure that while a caption is being generated, frames continue to be captured to avoid losing information. The first thread is responsible for capturing frames and displaying the images and captions on the screen. The second thread processes sets of frames to produce the captions using our LVC model. This caption generation thread can be parameterized to define both the parameter  $\Delta t$ , i.e., the length of the minimum video segment to be analyzed, and a memory parameter  $M$  that is maximum number of segments with length  $\Delta t$  that we will keep in memory to produce captions in the demonstrator. With this memory, we ensure that the dense caption prediction LVC module can access more context, a larger portion of the video, without losing its essence as a live system, and generate more accurate captions. Figure 15 provides a graphical description of the implementation made for the demonstrator.

As for processing speed, in Table 7, we report the average frames per second that our implementation is capable of processing and the average time it takes to generate a caption prediction by our LVC implementation. The camera we are working with



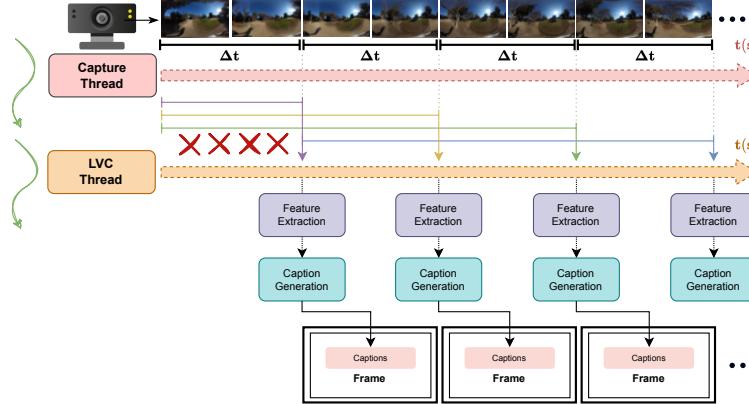
**Fig. 14 Bad Quality Example:** The predictions do not resemble the annotations in the video, and the proposed metric reflects the system’s failed behavior in an online mode.

provides a frame rate of 30 frames per second, so we can calculate the optimal length of the video segments to be used by the LVC system. We define  $l$  as the length in frames of the input video clip:  $l = 2.38 \cdot 30 \approx 71$  If the video segments we introduce to the model have a duration of 71 frames, we will ensure that when the processing of one clip ends, the next one is introduced into the LCV model. This way, no frames are discarded. All these tests for the demo were performed on a laptop running Ubuntu 18.04 operating system, equipped with an Intel Core i7 processor, and an integrated NVIDIA Quadro RTX 5000 MaxQ graphics card. Note that higher speed can be achieved if a more powerful GPU is used.

	FPS interfaz	Time of prediction (s)
Average	18.93	2.38

**Table 7** Number of frames per second (FPS) and average prediction time reported by our LVC demo system.

For the LVC problem, it is crucial to assess the performance of our model on devices with limited computational capacity, such as those used in embedded systems, robotics, IoT, or Edge Computing. To this end, we evaluated our LVC model on an



**Fig. 15** Multiprocessing demonstration scheme implemented. Our solution is capable of continuously displaying and producing captions. The system depicted in this figure employs a memory parameter  $M = 3$ , so that the latest caption generation only receives the 2 previous segments and the current one.

NVIDIA Jetson Xavier-NX board, a platform equipped with 384-core NVIDIA Volta GPUs and 48 Tensor Cores, capable of operating with a power consumption as low as 20W. This setup allowed us to analyze the model’s suitability for resource-constrained environments. To conduct these performance tests, we adapted the developed demo to operate efficiently on the NVIDIA Jetson Xavier-NX. First, we compiled the necessary libraries specifically for optimal use of the ARM64 architecture and the integrated GPU of the Jetson Xavier-NX board. After embedding the application into the platform, we analyzed the system’s performance, adjusting the parameter  $\Delta t$  as shown in Table 8. One key observation is that our implementation achieves a frame rate exceeding 0.8 FPS, which could be sufficient for many applications requiring an embedded LVC system. Notably, the maximum  $\Delta t$  value that the Jetson platform can process is 70 frames. It is worth highlighting that the processing of TSP features represents the main bottleneck. In percentage terms, the time spent generating these features consistently exceeds the time required for generating dense captions and filtering when  $\Delta t > 16$  frames.

$\Delta t$	FPS	Time of features (sec.)	Time of LVC (sec.)
16	0.87	0.39	0.75
48	0.51	1.15	0.75
70	0.11	7.6	2.4

**Table 8** Performance evaluation of our LVC model on an NVIDIA Jetson Xavier-NX board. We report: number of frames per second (FPS), and average time for the computation of features and captions.

## 5 Conclusions

Live video captioning is a novel and challenging problem that has not been deeply investigated in the scientific literature. As we have shown, generating dense captions for live video streams is a much harder problem than one might conclude from results reported in previous works under more constrained settings, e.g., offline dense video captioning models. In fact, traditional evaluation metrics need to be updated to novel online versions that allow us to judge the actual live performance of the LVC models. In this work, we have formalized the problem of LVC for the first time, proposed new metrics tailored to it. We introduced an LVC model capable of integrating transformer-based attention mechanisms with a caption filtering module for video streams received as input. Results of our model as well as an evaluation kit with the novel metrics integrated are made publicly available to encourage further research on LVC on realistic data: <https://github.com/gramuah/lvc>. We hope to encourage more researchers to look into the challenging yet very practical task of LVC. This work enhances our understanding of LVC and paves the way for live video understanding and accessibility applications in dynamic environments.

As future work, we consider the following options. The first would be to allow our LVC model to improve the predictions of the captions, using a memory mechanism, where the model takes into account past predictions. Technically, for example, recurrent networks could be incorporated into the model. A second line of improvement would consist of incorporating explainability techniques to improve the interpretability of our LVC model. Inspired by advancements in Explainable Artificial Intelligence (XAI) [44], such as attention visualization and feature importance analysis, we aim to provide insights into the model’s decision-making process. These methods could help bridge the gap between black-box models and transparent systems, enhancing trust and usability in applications requiring streaming video analysis. Finally, the future capabilities LVC systems are closely tied to advancements in sensor technology and video analysis techniques. Improvements in camera sensors, such as higher frame rates, and low-light performance, can provide richer data for more accurate caption generation. Additionally, emerging video analysis techniques, including multimodal feature fusion and lightweight neural architectures, may enhance the efficiency and robustness of LVC systems, particularly for real-time applications. While these developments are beyond the scope of this study, they represent promising directions for expanding the applicability and effectiveness of LVC models in diverse scenarios, including robotics, surveillance, and assistive technologies.

## Declarations

- Acknowledgements: We appreciate the assistance of Walfrido González Molina in generating some figures.
- Funding: This research was partially funded by projects: NAVISOCIAL, with reference 2023/00405/001 from the University of Alcalá; NAVIGATOR-D, with reference PID2023-148310OB-I00 from the Ministry of Science and Innovation of Spain.
- Conflict of interest/Competing interests: The authors certify that they have no conflict of interest.

- Ethics approval and consent to participate: Not applicable.
- Consent for publication: Not applicable.
- Data availability: The dataset used for all the experiments in this paper is *ActivityNet Captions* [13], which can be accessed at <https://cs.stanford.edu/people/ranjaykrishna/densevid/>.
- Materials availability: Not applicable.
- Code availability: Code to reproduce our results can be found at <https://github.com/gramuah/lvc>.
- Author contribution: Conceptualization, all; methodology, R.J.L.-S. and E. B.-F.; software, R.J.L.-S., E. B.-F., N.N. and C.G.-Á.; validation, all; ; writing—original draft preparation, all; supervision, R.J.L.-S. and S. M.-B.; project administration, R.J.L.-S. and S. M.-B.; funding acquisition, R.J.L.-S. and S. M.-B..

## References

- [1] Vaishnavi, J., Narmatha, V.: Video captioning – a survey. *Multimedia Tools and Applications*, 1–32 (2024)
- [2] Babavalian, M.R., Kiani, K.: Video captioning using transformer-based GAN. *Multimedia Tools and Applications*, 1–23 (2024)
- [3] Liu, S., Li, A., Zhao, Y., Wang, J., Wang, Y.: EvCap: Element-aware video captioning. *IEEE Transactions on Circuits and Systems for Video Technology* **34**(10), 9718–9731 (2024)
- [4] Yao, X., Zeng, Y., Gu, M., Yuan, R., Li, J., Ge, J.: Multi-level video captioning method based on semantic space. *Multimedia Tools and Applications* **83**(28), 72113–72130 (2024)
- [5] Tang, M., Wang, Z., Zeng, Z., Li, X., Zhou, L.: Stay in grid: Improving video captioning via fully grid-level representation. *IEEE Transactions on Circuits and Systems for Video Technology* **33**(7), 3319–3332 (2023)
- [6] Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pp. 5288–5296 (2016)
- [7] Wang, X., Wu, J., Chen, J., Li, L., Wang, Y.-F., Wang, W.Y.: Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4580–4590 (2019)
- [8] Artham, S., Shaikh, S.H.: A neural ODE and transformer-based model for temporal understanding and dense video captioning. *Multimedia Tools and Applications* **83**(23), 64037–64056 (2024)
- [9] Yang, A., Nagrani, A., Seo, P.H., Miech, A., Pont-Tuset, J., Laptev, I., Sivic, J.,

Schmid, C.: Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 10714–10726 (2023)

[10] Wang, T., Zhang, R., Lu, Z., Zheng, F., Cheng, R., Luo, P.: End-to-end dense video captioning with parallel decoding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 6847–6857 (2021)

[11] Zhang, Z., Xu, D., Ouyang, W., Tan, C.: Show, tell and summarize: Dense video captioning using visual cue aided sentence summarization. *IEEE Transactions on Circuits and Systems for Video Technology* **30**(9), 3130–3139 (2020) <https://doi.org/10.1109/TCSVT.2019.2936526>

[12] Wang, T., Zheng, H., Yu, M., Tian, Q., Hu, H.: Event-centric hierarchical representation for dense video captioning. *IEEE Transactions on Circuits and Systems for Video Technology* **31**(5), 1890–1900 (2021)

[13] Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C.: Dense-captioning events in videos. In: International Conference on Computer Vision (ICCV) (2017)

[14] Zhou, L., Zhou, Y., Corso, J.J., Socher, R., Xiong, C.: End-to-end dense video captioning with masked transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 8739–8748 (2018)

[15] Wang, J., Jiang, W., Ma, L., Liu, W., Xu, Y.: Bidirectional attentive fusion with context gating for dense video captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

[16] Yang, D., Yuan, C.: Hierarchical context encoding for events captioning in videos. In: 25th IEEE International Conference on Image Processing (ICIP), pp. 1288–1292 (2018)

[17] Iashin, V., Rahtu, E.: Multi-modal dense video captioning. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2020)

[18] Li, Y., Yao, T., Pan, Y., Chao, H., Mei, T.: Jointly localizing and describing events for dense video captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018). <https://doi.org/10.1109/CVPR.2018.00782>

[19] Zhou, L., Zhou, Y., Corso, J.J., Socher, R., Xiong, C.: End-to-end dense video captioning with masked transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8739–8748 (2018)

[20] De Geest, R., Gavves, E., Ghodrati, A., Li, C. Z. and Snoek, Tuytelaars, T.: Online action detection. In: European Conference on Computer Vision (ECCV) (2016)

- [21] Gao, J., Yang, Z., Nevatia, R.: RED: Reinforced encoder-decoder networks for action anticipation. In: British Machine Vision Conference (BMVC) (2017)
- [22] De Geest, R., Tuytelaars, T.: Modeling temporal structure with LSTM for online action detection. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1549–1557 (2018)
- [23] Li, Y., Lan, C., Xing, J., Zeng, W., Yuan, C., Liu, J.: Online human action detection using joint classification-regression recurrent neural networks. In: European Conference on Computer Vision (ECCV) (2016)
- [24] Baptista-Ríos, M., López-Sastre, R.J., Caba Heilbron, F., Van Gemert, J.C., Acevedo-Rodríguez, F.J., Maldonado-Bascón, S.: Rethinking online action detection in untrimmed videos: A novel online evaluation protocol. *IEEE Access* **8**, 5139–5146 (2020)
- [25] López-Sastre, R.J., Baptista-Ríos, M., Acevedo-Rodríguez, F.J., Martín-Martín, P., Maldonado-Bascón, S.: Live video action recognition from unsupervised action proposals. In: 17th International Conference on Machine Vision and Applications (MVA) (2021)
- [26] Hu, X., Dai, J., Li, M., Peng, C., Li, Y., Du, S.: Online human action detection and anticipation in videos: A survey. *Neurocomputing* **491**, 395–413 (2022)
- [27] Hori, C., Hori, T., Le Roux, J.: Optimizing latency for online video captioning using audio-visualtransformers. In: Interspeech, pp. 586–590 (2021)
- [28] Hoai, M., Torre, F.: Max-margin early event detectors. In: Proceedings of the IEEE International Conference on Computer Vision (CVPR), pp. 2863–2870 (2012)
- [29] Huang, D., Yao, S., Wang, Y., De La Torre, F.: Sequential max-margin event detectors. In: European Conference on Computer Vision (ECCV), pp. 410–424 (2014)
- [30] Kong, Y., Kit, D., Fu, Y.: A discriminative model with multiple temporal scales for action prediction. In: European Conference on Computer Vision (ECCV), pp. 596–611 (2014)
- [31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: NeurIPS, vol. 30 (2017)
- [32] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. In: Proceedings of the International Conference on Learning Representations (ICLR) (2020)
- [33] Suin, M., Rajagopalan, A.: An efficient framework for dense video captioning.

In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12039–12046 (2020). <https://doi.org/10.1609/aaai.v34i07.6881>

[34] Alwassel, H., Giancola, S., Ghanem, B.: Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops (2021)

[35] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression . In: Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 658–666 (2019)

[36] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal Loss for Dense Object Detection . In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2999–3007 (2017)

[37] Denkowski, M., Lavie, A.: Meteor universal: Language specific translation evaluation for any target language. In: Proceedings of the Ninth Workshop on Statistical Machine Translation, pp. 376–380 (2014)

[38] Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318 (2002). <https://doi.org/10.3115/1073083.1073135>

[39] Lin, C.-Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)

[40] Fabian Caba Heilbron, B.G. Victor Escorcia, Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–970 (2015)

[41] Ghanem, B., Niebles, J.C., Snoek, C., Caba Heilbron, F., Alwassel, H., Escorcia, V., Khrisna, R., Buch, S., Duc Dao, C.: The ActivityNet Large-Scale Activity Recognition Challenge 2018 Summary (2018)

[42] Xiong, Y., Dai, B., Lin, D.: Move forward and tell: A progressive generator of video descriptions. In: European Conference on Computer Vision (ECCV), pp. 489–505 (2018)

[43] Mun, J., Yang, L., Ren, Z., Xu, N., Han, B.: Streamlined dense video captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

[44] Akkem, Y., Biswas, S.K., Varanasi, A.: Streamlit-based enhancing crop recommendation systems with advanced explainable artificial intelligence for smart

