



# Finding epic moments in live content through deep learning on collective decisions

Hyeonho Song<sup>1,2</sup>, Kunwoo Park<sup>3\*</sup>  and Meeyoung Cha<sup>2,1\*</sup> 

\*Correspondence:

[kunwoo.park@ssu.ac.kr](mailto:kunwoo.park@ssu.ac.kr);  
[mcha@ibs.re.kr](mailto:mcha@ibs.re.kr)

<sup>2</sup>Data Science Group, Institute of Basic Science, Daejeon, Republic of Korea

<sup>3</sup>School of AI Convergence, Soongsil University, Seoul, Republic of Korea

Full list of author information is available at the end of the article

## Abstract

Live streaming services enable the audience to interact with one another and the streamer over live content. The surging popularity of live streaming platforms has created a competitive environment. To retain existing viewers and attract newcomers, streamers and fans often create a well-condensed summary of the streamed content. However, this process is manual and costly due to the length of online live streaming events. The current study identifies enjoyable moments in user-generated live video content by examining the audiences' collective evaluation of its epicness. We characterize what features "epic" moments and present a deep learning model to extract them based on analyzing two million user-recommended clips and the associated chat conversations. The evaluation shows that our data-driven approach can identify epic moments from user-generated streamed content that cover various contexts (e.g., victory, funny, awkward, embarrassing). Our user study further demonstrates that the proposed automatic model performs comparably to expert suggestions. We discuss implications of the collective decision-driven extraction in identifying diverse epic moments in a scalable way.

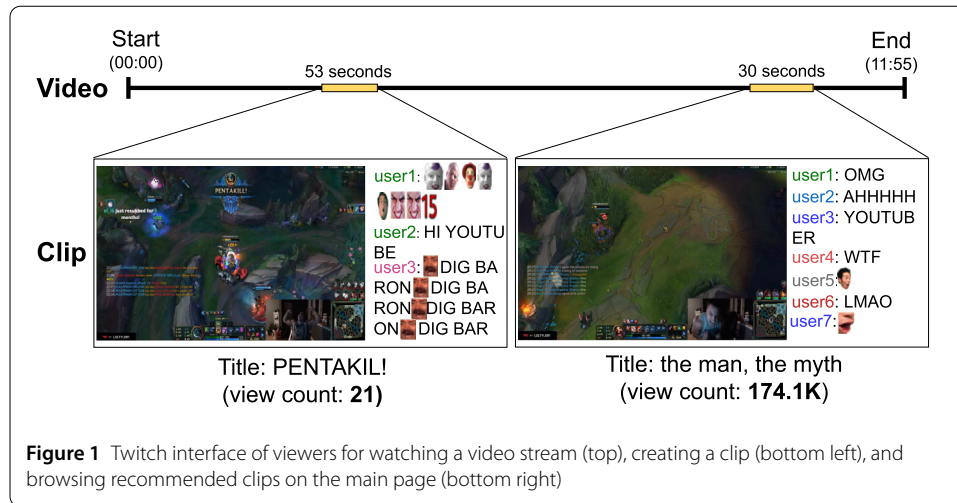
**Keywords:** Epic moment; Online live streams; Crowdsourced decisions; Twitch.tv

## 1 Introduction

Live streaming has emerged as a central media consumption behavior. Unlike traditional media channels, modern live streaming platforms enable viewers to interact with one another and on-air broadcasters through live chats. The combination of high interactivity and engaged audiences offers new opportunities for broadcasters. For example, fandom networks have become a widespread social phenomenon, and interesting live-streamed content can be re-packaged through various channels. Platforms like Twitch host a wide range of content creators; ordinary Internet users can broadcast their gameplays and hobbies alongside experts. Twitch, as of June 2021, hosts 9.3 million monthly broadcasters and 2.9 million concurrent viewers [1].

The increasing amount of live streaming content and broadcasters provides valuable opportunities as a democratic media platform introduces challenges in content promotion and searching. As a result, streamers need to employ various strategies to draw new viewers in the competitive environment and keep their current viewers engaged. One such strategy is to create a shorter version of a minute-sized streaming video as *epic moments*

© The Author(s) 2021. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.



that are well-condensed short clips. These epic moments function as an enjoyable bite-sized summary of long video content, generated typically from several hours of content. Short clips can also be used to promote the streamer's channel across other web platforms, such as Reddit and YouTube. Epic moments are similar to video highlights in that they are both short summaries of long videos, yet the two are defined differently. Epic moments represent "enjoyable" moments [2–4], whereas highlights are "informative" in nature [5–8].

Despite the potential use of epic moments, the research community and industry do not have a systematic method to identify them from hours of seemingly mundane user-generated video streams. Most streamers and channel managers (who are now in the order of millions) manually spot informative moments and edit them. This heuristic process is costly because manual inspection requires tremendous time and effort. The current paper asks whether the multi-modal information of visual signals and audience reactions can help build a machine learning model for extracting epic moments.

This paper characterizes the audience's collective dynamics on a user-generated online streaming service and proposes a method to identify well-condensed and entertaining clips representing epic moments. Our data-driven approach assumes that user reactions embedded in the live chat conversations will reveal markers of key moments that likely appeal to the audience. Rather than naive metrics like chat frequency, we aim to utilize the emotional reactions that are both verbally and non-verbally expressed in chats via a neural network model.

To understand what comprises epic moments in user-generated online streaming systems, we examine Twitch's existing clip collection. Anyone can generate clips and share them on social media. Figure 1 shows two example clips created by different viewers from an identical stream. The clips have different lengths and contexts (e.g., casual, victory). These bite-sized user-defined clips are easily shared on external sites, acting as promotional content for the video channel. They also serve as a valuable resource to examine collective audience dynamics that reveal what kinds of content are likely re-packaged and how popular clips of different contexts become.

For the extensive collection of clips,<sup>1</sup> we also gathered the live chat conversations that occurred during each streamed video to analyze the audience's collective behaviors. Knowing which factors are associated with popular clips helps advance our understanding of epic moments in user-generated streams, further enabling us to build AI-driven approaches to suggest them automatically.

The main findings and contributions of this study are threefold:

1. From chat logs from live streams, we classify characteristics of popular moments based on three features: (i) audience engagement, (ii) chat content, and (iii) audience sentiment. We model the audience sentiment by employing an embedding approach representing the meanings of emoting expressions in a fixed-dimensional vector space. We release the pre-trained embedding vectors of Twitch emotes for the research community.
2. We present a deep learning model that detects enjoyable moments by classifying popular and ordinary moments. This model is interpretable and describes which data modality is important for the decision. Experiments confirm that the model can detect popular clips with an accuracy of 0.8261 and that the combined usage of multimodal features (i.e., chat and video content) is critical.
3. We conduct a user study to evaluate our algorithm-based suggestions against the human alternative (i.e., expert-edited epic moments). The crowdsourced experiment reveals the potential that the algorithmic suggestions are comparable to expensive human decisions. Furthermore, algorithmic suggestions cover diverse emotions, such as funny, sad, exciting, and nervous, as intended.

## 2 Related work

### 2.1 Chat dynamics in live stream services

Many live streaming platforms allow their audience to interact concurrently with the streamed content. As the audience converse themselves and share a common interest in the content, social dynamics emerge. Previous research on live streaming services has investigated such phenomenon from different perspectives.

Some studies emphasize the importance and uniqueness of chat in live streaming. They find engagement via chat to be an important streaming service factor since it strengthens bonding between the broadcaster and viewers [9], which itself is gratifications [10]. For instance, users build a parasocial relationship with streamers and other viewers [11] and form instant social communities [12]. Viewers also build close relationships through interaction via chat [13]. The anonymity in online conversations allows users to express themselves quickly based on shared interests. These functions reinforce the viewers to revisit the video channel.


This conversation is temporal in that only the recent chats are displayed on the screen. The single-screen makes the conversation involving numerous users challenging to follow and is called the "waterfall of the text" [12]. A study compared the characteristics of chat dynamics involving numerous users with small-scale conversations [14]. They found that large-scale conversations tend to contain chat messages that are shorter and less original. Another study identified that the increased complexity of chat conversations can induce a cacophony, in which the amount of information and length in messages tends to

---

<sup>1</sup>We release the dataset as supplementary information within the scope of Twitch's terms of use.

decrease [15]. One study also found that conversation patterns can reveal cues for understanding the main content [16]. A large-scale conversation with high frequencies is associated with major events in a stream. On the other hand, viewers post meaningless messages of emotes in boring scenes. Another study suggests that both sentiments of broadcasters and viewers are affected by each other [17]. For instance, chats with positive sentiment can cause a streamer's positive verbal expression and vice versa.

The research attempted to characterize chat contents and how they differ based on streamer and audience types. One study classified Twitch users into four clusters: chatbot, subscriber, regular user, and turbo users [18]. Using a regression model, they found that Twitch users tend to imitate the previous messages. Users with higher authority had more impact on making other users copy their messages. Another study showed that audience participation and streamer performance vary by the audience's scale through a clustering analysis [19]. On the other hand, a study examined how chat content differs according to the streamer's gender using a word embedding method [20]. Results reveal that gendered conversations are prevalent in the Twitch community. Similarly, using manual annotations, other researchers found that female streamers tend to receive sexual comments more frequently than are male streamers [21].

Research on Twitch has focused on emote, a mode of communication expressing emotion through a tiny image. On Twitch, platform-specific emotes are used more widely and frequently than emojis, a general set of ideograms and facial expressions used across diverse online platforms. There has been a line of studies examining emojis. A notable study is emoji2vec [22], which achieved embedding vectors by learning textual expressions corresponding to each emoji. However, it cannot be easily transferred for Twitch emotes, which lack an agreed definition. For instance, Kappa () is the gray-scale face of a former Twitch employee, which was initially included as an Easter egg but has been exploited to represent sarcasm by Twitch users. A study employed word2vec [23] to extract chat logs' representation to predict the audience sentiment [24]. Compared to a dictionary-based method [25], it predicted audience sentiment with a better performance.

This study examines the difference between popular and ordinary moments in terms of diverse chat characteristics. Also, we present a method of embedding Twitch emotes' meaning from the context of emote expressions.

## 2.2 Machine learning for highlight extraction

Previous research has tried to develop machine learning methods for generating highlights of various types of content and considered diverse types of online contents: music [26], sports [27], and e-sports [5, 28, 29], to list a few. These studies can be classified into two categories by the employed type of machine learning method: supervised and unsupervised approaches.

Supervised approaches extract features from highlight videos crafted by experts and learn to discriminate their patterns from ordinary moments. A study presented a support vector machine model for generating highlights of e-sport matches [29]. Another study proposed a latent linear ranking model based on a support vector machine (SVM) for extracting video highlights by learning on highlight index defined by pairwise ranking, which was derived from YouTube videos crafted by editors [30]. Some studies measure the user's arousal as a proxy of highlight and introduce machine learning techniques with video data or game-level log data to predict them [31, 32]

Another line of research employed unsupervised approaches for learning to detect highlight videos. Unlike supervised approaches that require pre-edited highlight videos of domain experts, the unsupervised approaches extract highlight moments based on a heuristic. A study identified music highlights by assuming common patterns of highlights across songs of the same genre [26]. One study assumed that extracting high viewership moments are highlights, and they proposed an embedding model that represents words and emotions in the same space for prediction [28].

Unlike previous approaches, we propose to employ user-recommended short videos and their corresponding view counts for detecting enjoyable moments and combining them for recommending epic moments, which is a collection of the most enjoyable moments.

### 3 Data methodology

#### 3.1 Background: twitch.tv and user-generated clips

Twitch (<http://www.twitch.tv/>) is one of the most popular live streaming platforms. According to alexa.com, the service is ranked 33 in the list of top sites representing the most popular website globally (at the time of submission). *Streamers* broadcast live videos and own channels, and *viewers* are those who consume streaming content. Popular streams can reach up to more than thousands of concurrent viewers. In Twitch chat conversations, viewers actively use Twitch-specific emojis called *emotes* to express diverse types of reactions with just a few bytes. Twitch specializes in live streaming on video games. Individual streamers broadcast their gameplay to show off their competence or introduce new games, and companies stream e-sports matches of professional players. This study focuses on individual streaming because e-sport matches lack interaction between a broadcaster and viewers.

Figure 1 shows an example of the viewer's interfaces on Twitch. In addition to watching a stream and engaging in a chat conversation, Twitch enables viewers to manually capture a short segment of a video and share it as a 'clip.' The length of a clip can vary from 5 to 60 seconds. Once a clip is generated, it is shown on the channel page. Some clips appear on the Twitch main page based on recommendation engines and get shared on external websites via URLs.

Twitch keeps track of view counts for each clip, which is a measure of popularity based on collective audience reaction. This study assumes that clips with high view counts can be candidates for epic moments whose characteristics are distinct from ordinary moments. In Sect. 4, we analyze the diverse characteristics of clips and view counts for a systemic understanding of such differences. Section 5 proposes a model that learns important cues of popular moments from video and chat data.

#### 3.2 Data collection

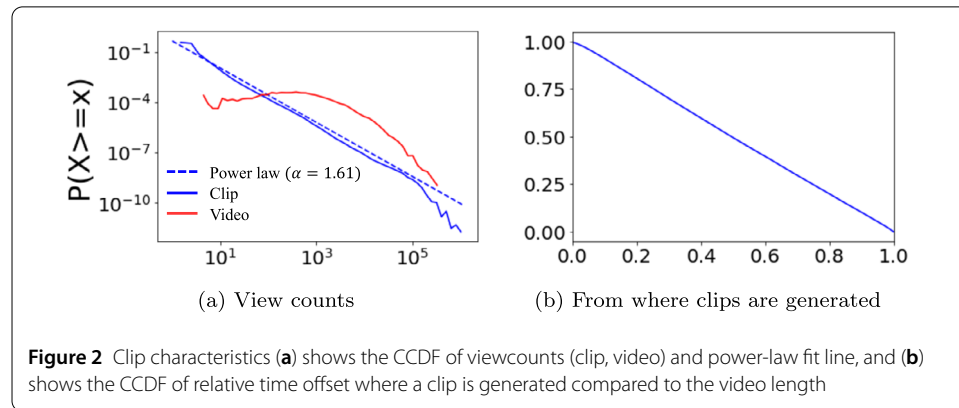
We first compile a list of popular streamers from two websites that provide statistics on streamers [33, 34]. We exclude non-English, e-sports, and promotional channels. Considering the low proportion of female streamers and potential bias in audience reactions as reported in [20, 21], we further exclude female streamers (21 of them in total) in the final analysis. For the remaining 213 male streamers,<sup>2</sup> using the Twitch API, we collected videos of recorded video streams, chat conversations, and clips that were shared from May

---

<sup>2</sup>The average number of the viewer and follower per channel is 50,965,700.5 and 1,012,030.6, respectively.

**Table 1** Descriptive statistics of the collected dataset

Type	N	Variable	Mean	Median	Min	Max
Video	11,779	Clip count	187.90	110	1	1676
		Utterance count	19,842.31	9396	2	459,650
		Viewcount	5137.25	1928	0	360,636
		Duration (sec)	20,233.63	19,413	8	172,807
Clip	2,213,247	Utterance count	71.53	36	1	820
		Viewcount	78.25	2	1	1,122,235
		Duration (sec)	29.77	30	5	60



**Figure 2** Clip characteristics (a) shows the CCDF of viewcounts (clip, video) and power-law fit line, and (b) shows the CCDF of relative time offset where a clip is generated compared to the video length

to June 2019. To promote future studies on gender bias, we release the emote embedding results trained on audience chats of the studied male-only dataset as well as the dataset including both genders.

Table 1 presents descriptive data statistics. For videos, the average stream length is approximately 5.6 hours, which is typical of gameplays. The average number of utterances (i.e., chat sentences) is 19,842.31, representing active audience interaction. We make two observations from the gathered data.

First, as shown in Fig. 2(a), the view counts of clips and videos are highly skewed. The clip view count follows a power-law distribution and the blue dashed line of the alpha of 1.61. According to the Shapiro–Wilk test on the log-transformed video view count, there was not enough evidence for rejecting the null hypothesis on normality ( $W = 0.999$ ,  $p$ -value = 0.088). This finding suggests a log-normal distribution.<sup>3</sup> A skewed distribution of clip view counts implies that only a few clips can eventually become popular.

Second, Fig. 2(b) shows the complementary cumulative distribution function (CCDF) on the relative time offset of generated clips compared to the entire length of a video. It approximates a linear function, suggesting that a clip is equally likely to be generated over different video stages. The time position within the live stream does not provide much meaningful information about which video segment will later be chosen as an epic moment.

#### 4 Chat characteristics of popular clips

By utilizing view counts on clips, we may examine what kinds of the live streaming segment is considered more popular. We used view counts to identify three ordinal groups,

<sup>3</sup>Since it is not allowed to use more than 5000 data points for the test, we repeated the test with sampling 5000 instances 100 times and reported the average.

**Table 2** Label distribution

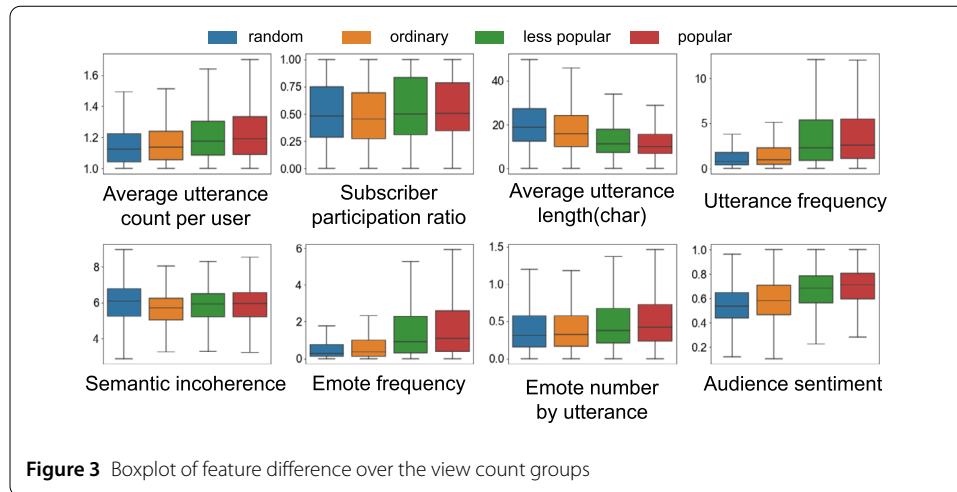
Type	Count
Popular	105,741
Less popular	592,639
Ordinary	1,514,867
Random	2,213,247

considering its skewed distribution: popular ( $\geq 34$ , clips with top 5% of view counts), ordinary ( $\leq 2$ , bottom 50%), and less popular ( $< 34$  and  $> 2$ ). We also chose a random segment of the original live content and turned considered them as ‘random’ clips as a reference point. We set the duration of random moments as 30 seconds to match the average clip duration. The label distribution is presented in Table 2.

#### 4.1 Feature comparisons

For comparison between the four clip groups, we measure features related to chat frequency and viewer engagements based on the literature of chat dynamics [14, 35] and user loyalty [36, 37] in live stream services.

- Utterance frequency & the average utterance length: We measure the scale of audience chat in two ways; First, we divided the number of utterances by clip duration (seconds) for the clip’s utterance frequency. Second, for the average utterance length, we divided the total number of characters by the number of utterances.
- Emote frequency & emote frequency by utterance: To quantify usages of emote expressions, we measure the emote frequency, which is the number of emotes in a clip divided by the corresponding length. An average number of emotes per utterance is calculated for capturing emote usages of distinct users.
- Subscriber participation ratio & the average utterance count per user: To approximate user loyalty, we examine the fraction of channel subscribers among those who participated in chats. Twitch users can support channels via subscription, which requires a monthly fee (\$4.99). Subscribers receive channel-specific emotes and badges as well as ad-free viewing. Thus it can be considered a proxy of loyalty. Also, we measure the average degree of participation by calculating the average utterance count per user because user participation is an important indicator of popularity [38].
- Semantic incoherence: Previous research noted Twitch audience chats tend to become semantically coherent in certain moments [14]. We assume that Twitch audiences may exhibit semantically coherent messages in a popular moment. To quantify the (in-)coherence, we calculated Shannon’s entropy based on a bag of word model. We first construct a unigram dictionary that appears more than 100 times. We then calculate each word’s probability by dividing the number of each word occurrence by the total number of words in a whole set of utterances in the audience chat of a given moment. The Shannon entropy is measured on the frequency vector of each corresponding chat. We call the measure semantic incoherence. As entropy is the highest for a uniform distribution, a low semantic incoherence value indicates that the audience tends to use similar word expressions.
- Audience sentiment: The audience may exhibit a large amount of sentiment according to the context of a live stream, and we assume that a popular moment may contain more sentiment in chats. To quantify the degree of audience sentiment in a given moment, we measured the sum of positive and negative sentiment values using the



**Table 3** Rank correlation of each feature against the view count group measured by Spearman’s  $\rho$ . Meaningful effect sizes are marked as bold. All correlations are statistically significant ( $p < 0.001$ )

Variable	Rank correlation
Audience sentiment	<b>0.2537</b>
Utterance frequency	<b>0.2394</b>
Emote frequency	<b>0.2101</b>
Audience utterance count per user	0.1144
Number of emote by utterance	0.0651
Subscriber participation ratio	0.0052
Semantic incoherence	-0.1209
Average utterance length	<b>-0.2406</b>

valence aware dictionary and sentiment reasoner (VADER) [25]. This represents the normalized sentiment intensity that is expressed in a chat (0~1).

Figure 3 demonstrates the difference in features between the four groups on view count. Conducting a one-way analysis of variance (ANOVA) test on each feature, we found that each group’s features were significantly different ( $p < 0.001$ ). The Tukey honestly significant difference test (Tukey HSD) confirms that there exist statistically significant differences for every pair of popularity groups for each feature ( $p < 0.001$ ). To determine whether a feature is correlated against the view count, we measure Spearman’s  $\rho$  of each feature against the ordinal variable of the clip’s view count: random (0), ordinary (1), less popular (2), and popular (3). As shown in Table 3, we discover the meaningful size of correlations for audience sentiment (0.2537), utterance frequency (0.2394), emote frequency (0.2101), and average utterance length (−0.2406).

The positive correlation of audience sentiment suggests that the audience expresses stronger emotion (positive or negative sentiments) in the popular clips. The correlations for utterance and emote frequencies suggest that emote usage increases for the popular clips. The high chat frequency in the popular clips might be related to the negative correlation of the average utterance length because a long message is challenging to be read in such “waterfall of the text” [12]. The observation of the negative correlation of semantic incoherence and clip popularity can be considered representative of coherent messages from popular clips, which can be interpreted as characteristic of crowdspeak [14]. At the same time, the positive correlation of semantic incoherence within the clips can be inter-



preted as the divergence of user expression, as described in [39]. However, we do not make a strong argument here as we do not investigate the causal connection.

## 4.2 Emote embedding analysis

Previous studies noted the importance of emote expressions as a major mode of communication on Twitch [28, 40]. To understand how the live-streaming audience expresses emotions differently during popular moments instead of ordinary moments, we propose using an embedding approach. Embedding was proposed initially to represent word semantics in a fixed-dimensional space. Here, the assumption is that emotes appearing together with a particular set of words in the same utterance convey similar emotions.

To train the emote embedding model, we first constructed two corpora from our dataset: *Emote-only* and *Whole*. The emote-only corpus contains emote identifiers, and the whole corpus consists of emote identifiers and word tokens. For both corpora, we concatenate all chat utterances in a video after necessary preprocessing steps: cleaning characters repeated more than twice (e.g., 'gooood' becomes 'good') and removing user mentions (e.g., '@username') as well as URLs. We consider emote identifiers that are provided by Twitch and popular third-party extensions<sup>4</sup> for analysis.

To facilitate learning emote representation, we consider word tokens appearing together with emotes in chat conversations. We apply the continuous bag-of-words (CBOW) model of word2vec [23] using the whole corpus and emote-only corpus separately. After that, we fine-tune the entire corpus model by replacing the emote embedding vectors with those trained on the emote-only corpus. We set the embedding dimension to 100, the minimum occurrence of tokens to 100, and the window size to 5. This sequential learning procedure efficiently captures the meaning of emote expressions based on the word embedding algorithm effectiveness, as identified in diverse applications [41, 42]. We tested the skip-gram model, but it achieved worse performance than the CBOW model.

Next, we identify emote clusters based on the learned embedding representation. To find a set of emotes of which meanings are closely related to one another, we utilize the density-based spatial clustering of applications with noise (DBSCAN) algorithm. For training, we set the epsilon value to 0.49 and the number of minimum samples to 12. For comparison, the baseline embedding model is trained on the entire corpus without any fine-tuning step to be consistent with the previous study [24].

Table 4 presents the clustering results evaluated by the silhouette coefficient and Davies–Bouldin index. The first metric is the silhouette coefficient, which measures within-cluster elements' average distance and the minimum distance to other cluster elements for each item. The other measure is the Davies–Bouldin index, which is based on Euclidean distances. Davies–Bouldin index quantifies the degree of dispersion within the cluster by calculating the distance between the element and its center. The higher silhouette coefficient and the lower Davies–Bouldin index indicate a better set of clusters. Our method achieves a baseline approach in both measures, confirming fine-tuning embedding using the emote-only corpus contributes to meaningful representation.

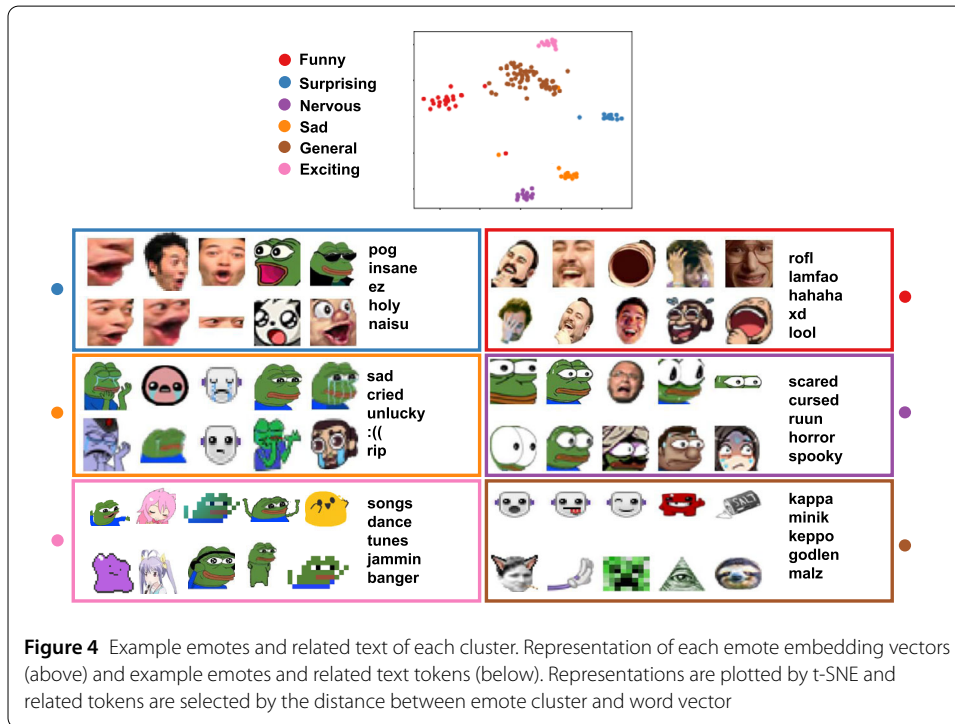
Figure 4 illustrates the clustering results on the two-dimensional space identified by the t-distributed stochastic neighbor embedding (t-SNE). The color indicates a cluster's membership, and the plot presents five-word tokens closest to each cluster. Results reveal that

---

<sup>4</sup>BetterTTV and FrankerFaceZ.

**Table 4** Evaluation of clustering results

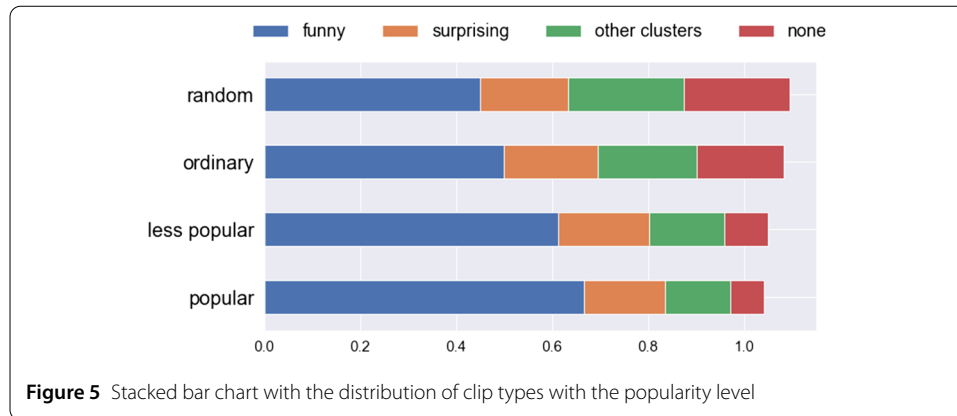
Measure	Our model	Baseline
Silhouette Coefficient	0.351	0.047
Davies–Bouldin index	0.411	0.829



each identified cluster contains similar-looking emotes even though a visual feature has not been incorporated. The representative word tokens for each cluster suggest that the clustered emote expressions are related to the emotional expressions of the text. For example, the orange-colored cluster tends to contain emotes of crying faces (e.g., 🥺), which is closely located in the embedding space to the corresponding words such as ‘sad’ and ‘cried.’ Based on this observation, we name each cluster based on the word tokens corresponding to each cluster as funny, surprising, nervous, sad, general, and exciting.

We investigate how the emote clusters appear in the audience chat against the clip’s view count based on the clustering results. The most frequent two groups are the funny and surprising clusters, which appear in 53.86 and 19.25 percent of clips, respectively. The nervous, sad, general, and exciting clusters only take up 6.38, 5.93, 4.00, and 2.63 percent, respectively. In case the audience chat message contained no emote at all or contained emote kinds that do not belong to any of the emotion clusters, we consider such chat messages as *none* type. The none type makes up 18.56% of the total clips. A clip can belong to one or more clusters if the emotes of the multiple clusters appear in the audience chat. This observation suggests that many of the emotes in clips are used to describe funny or surprising moments (or the same emotions at once).

To further understand the relationship between the emote expressions and the clip’s popularity, we examine the emote cluster distribution across the view counts group. Figure 5 shows the summed ratio of the emote clusters for the different view count groups.



The sum can be larger than 1.0 because clip types are inclusive (e.g., a clip can be funny and surprising). Here, we make several observations. The proportion of funny clips increases over view counts, i.e., funny moments may appear in all contexts, but the most frequently in the popular videos. On the other hand, the proportion of surprising clips is similar across the view counts groups. Second, the clips with a higher view count tend to contain the other sentiment-related emotive expressions less; Yet, it is noteworthy that more than 10 percent of popular clips still exist in which a diverse set of emotions are expressed. Lastly, the proportion of clips involving the other clusters and the none type decreases as the view count increases.

This observation suggests that the emotive expressions related to audience sentiment may convey implicit signals for popular moments. However, only relying on dominant emotions such as funny and surprising does not capture all of the epic moments in a Twitch video stream.

### 4.3 Summary of the findings

This section explored how the clip characteristics differ over the view counts or content popularity. Among the clip characteristics, we focused on various audience engagement levels and embedded sentiments seen from emotes. We found popular clips to contain stronger emotions, higher emote usage per utterance, at the same time, and shorter text length per utterance. Emote usage plays a critical role in Twitch because they allow users to express diverse emotions via short community-specific norms and images. To infer the meaning of emotes, we followed the existing literature methods and examined the embedding of emotes. We identified six clusters of emotes that occurred in similar contexts, which we termed as 'emote clusters.' They represent diverse audience reactions, including funny, sad, surprising, nervous, exciting, and general.

The joint analysis of clip groups and emote embedding reveals that the funny and surprising sentiments are two predominant emotions expressed by the audience on Twitch's live streaming content. The more popular a clip was, the higher the fraction of them contained these two emotions. However, a non-negligible (over 35%) fraction of the most popular clips were on different emotions. These findings indicate that popular clips, instead of ordinary clips or random moments, contain stronger audience reactions. It suggests the need to utilize a wide range of live-streamed content as clips beyond mere funny or surprising moments.

## 5 Popular moment detection

This section extends the previous findings on feature-level differences and constructs a machine learning model to detect live streaming segments that will likely become popular clips. Previous research defined a video highlight as a subset of the raw video that is “most informative” of its purpose (e.g., scoring moments in sports games) [5–8]. Another line of research considered identifying short clips that are of users’ interest [2–4]. We define an epic moment as a collection of the most enjoyable moments collectively determined by viewers out of a long video stream. Thus, the definition differs from video highlights in that epic moments may not necessarily be informative. It is also different from popular moments, which are consecutive video segments with high audience reactions. Rather epic moments can include diverse contexts, including failing or funny reactions as long as viewers collectively consider them enjoyable. Hence the epic moment extraction in this study consists of two steps: (i) identifying multiple enjoyable moments that are short and consecutive and (ii) combining them for representing the whole video. This section introduces the first step that predicts popular moments through deep learning on user-defined clips and their view counts.

### 5.1 Problem definition

We propose a detection method of popular moments based on the audience chat characteristics and meta-information about the streamed content. Formally, we address the classification task given a clip  $C$  as a tuple  $(V, T)$ , where  $V$  is a sequence of video frames during an arbitrary moment in a stream, and  $T$  is the accompanying audience chat conversations shared over the same period. Our goal is to determine a binary label  $Y$  reflecting that a clip is popular or not.  $Y$  is 1 when the view count is  $\geq 34$  (popular) and 0 when the view count is  $\leq 2$  (ordinary). We do not consider clips of moderate popularity to ensure efficient learning.

### 5.2 Model description

#### 5.2.1 Chat frequency-based model

Chat frequency was one of the distinguishing markers of popular clips in the previous section. As one of the baseline models, we present a neural network model that learns to classify clip popularity based on frequency features. We adopt a recurrent neural network (RNN) classifier, in which the base unit is the long short-term memory (LSTM) network. To model the chat frequency varying over time, we split the chat utterances shared over a clip period into multiple bins per every  $t$  second. For each bin, we measure the number of utterances ( $x_u$ ), the fraction of emotes ( $x_e$ ), the number of chat participants ( $x_p$ ), and the average number of lexicons per utterance ( $x_l$ ). To prevent the model from learning channel popularity, we normalize the chat frequency and the number of unique chat participants by the number of followers of the corresponding channel.

Formally, taking the vector  $\mathbf{v}_j$  for a given clip  $c$  as input, the chat frequency-based model outputs prediction regarding whether  $c$  is popular (1) or not popular (0):

$$\mathbf{v}_j = \begin{bmatrix} x_u^j \\ x_e^j \\ x_p^j \\ x_l^j \end{bmatrix}, \quad j = 1, 2, \dots, \left\lceil \frac{L_c}{t} \right\rceil,$$

$$\mathbf{c}_j, \mathbf{h}_j = \text{LSTM}(\mathbf{v}_j, \mathbf{c}_{j-1}, \mathbf{h}_{j-1}), \quad \mathbf{c}_j, \mathbf{h}_j \in \mathcal{R}^d, \tag{1}$$

$$\bar{\mathbf{h}}_{\text{freq}} = \frac{1}{\left\lceil \frac{L_c}{t} \right\rceil} \sum_{j=1}^{\left\lceil \frac{L_c}{t} \right\rceil} \mathbf{h}_j,$$

$$\hat{\mathbf{y}} = \text{argmax}(W_{\text{out}} \bar{\mathbf{h}}), \quad W_{\text{out}} \in \mathcal{R}^{2 \times d},$$

where  $L_c$  is the duration of  $c$ ,  $d$  is the dimension of the hidden and cell state, and  $t$  is the length of the bin in seconds.  $\mathbf{c}_j$  and  $\mathbf{h}_j$  are the cell and hidden state of LSTM at step  $j$ , respectively. The hidden states of LSTM are averaged and input into a linear transformation, followed by a sigmoid activation to make the binary prediction. We set the number of hidden layers in the LSTM to 2,  $d$  to 128 and  $t$  to 5.

### 5.2.2 Chat content-based model

Next, we utilize chat content in audience conversation. This model updates its features through a recurrent layer after transforming a word sequence into a sequence of embedded vectors. For each clip  $c$ , the word tokens in consecutive chat utterances  $T_c$  are transformed into one-hot vectors ( $\mathbf{w}_j, j = 1, 2, \dots, L$ , where  $L$  is the number of tokens in the clip period). Then, these word tokens are transformed into word embedding vectors ( $\mathbf{v}_j$ ); then, taking the vector  $\mathbf{v}_j$  of given a clip  $c$ , this model outputs a binary prediction regarding whether  $c$  is popular or not as follows:

$$\mathbf{v}_j = W_{\text{Emb}} \mathbf{w}_j, \quad W_{\text{Emb}} \in \mathcal{R}^{v \times e}, \mathbf{w}_j \in \mathcal{R}^v, \mathbf{v}_j \in \mathcal{R}^e,$$

$$\mathbf{c}_j, \mathbf{h}_j = \text{LSTM}(\mathbf{v}_j, \mathbf{c}_{j-1}, \mathbf{h}_{j-1}), \mathbf{c}_j, \mathbf{h}_j \in \mathcal{R}^d,$$

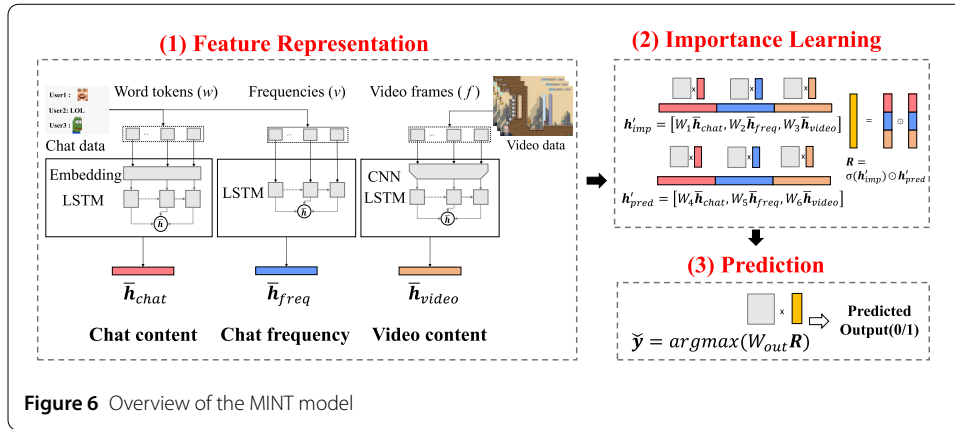
$$\bar{\mathbf{h}}_{\text{chat}} = \frac{1}{L} \sum_{j=1}^L \mathbf{h}_j, \tag{2}$$

$$\hat{\mathbf{y}} = \text{argmax}(W_{\text{out}} \bar{\mathbf{h}}_{\text{chat}}), \quad W_{\text{out}} \in \mathcal{R}^{2 \times d},$$

where  $W_{\text{Emb}}$  is the embedding matrix initialized by the CBOW model pretrained on the whole corpus, of which the details are present in Sect. 4. In addition,  $v$  is the size of vocabulary in the embedding model,  $e$  is the dimension of embedding vector, and  $d$  is the hidden and cell state dimension. We set the embedding vector dimension to 100 and the vocabulary size to 139,347. We set the number of the hidden layers in LSTM to 2 and  $d$  to 128.

### 5.2.3 Video content-based model

Following the previous practice on video highlight detection [5, 43], we build a deep learning classifier that learns visual features by incorporating a convolutional neural network



(CNN) with RNN. Given  $V$  comprising 60 frames per second, we feed every 60th frame into the CNN layer, and the recurrent layer is in charge of learning the dependency between the visual features. The model can be formally defined as follows:

$$\begin{aligned}
 \mathbf{v}_j &= \text{CNN}(\hat{\mathbf{x}}_j), \quad j = 1, 2, \dots, [L], \hat{\mathbf{x}}_j \in \mathcal{R}^{w \times h}, \mathbf{v}_j \in \mathcal{R}^v, \\
 \mathbf{c}_j, \mathbf{h}_j &= \text{LSTM}(\mathbf{v}_j, \mathbf{c}_{j-1}, \mathbf{h}_{j-1}), \quad \mathbf{c}_j, \mathbf{h}_j \in \mathcal{R}^d, \\
 \bar{\mathbf{h}}_{video} &= \frac{1}{L} \sum_{j=1}^{[L]} \mathbf{h}_j, \\
 \hat{\mathbf{y}} &= \text{argmax}(W_{out} \bar{\mathbf{h}}_{video}), \quad W_{out} \in \mathcal{R}^{2 \times d},
 \end{aligned} \tag{3}$$

where  $\hat{\mathbf{x}}_j$  is the resized input image,  $w$  and  $h$  are the width and height of the image, respectively,  $L$  is the clip duration in seconds, and  $v$  is the dimension of the CNN's hidden layer. As previous studies and practices adopted the residual network [5, 43] for its efficiency (i.e., not too large) and effectiveness (i.e., high performance), we utilize the residual network with 18 layers (ResNet-18 [44]), which were pretrained on ImageNet [45]. We resize each frame with the  $w$  of 224 and  $h$  of 224; accordingly,  $v$  becomes 1000. For the LSTM layer, we set the number of the hidden layers to 3 and the dimension to 128 ( $d = 128$ ).

#### 5.2.4 Multimodal detection with INTerpretability (MINT)

We propose a model named Multimodal Detection with INTerpretability (MINT) by merging each deep learning-based model (i.e., chat, video, frequency features). The comprehensive features from these three domains may capture different aspects of epic moments, and combining those cues leads to a better prediction. Here, we propose to combine them while jointly learning the importance of each modality.

We illustrate the model architecture in Fig. 6. The model tries to predict whether the given clip is popular or not following the three steps: (step-1) feature representation, (step-2) importance learning, and (step-3) prediction. The model represents chat content, chat frequency, and video content in the feature representation step using the same architecture as the model corresponding to each modality. In the second step, it combines hidden units of different modalities by inferring the importance. We add two different  $n$ -to-1 transformation layers to each modality. The first set of transformation ( $W_1, W_2, W_3$ ) computes a score from each modality, and the second set of transformation

( $W_4, W_5, W_6$ ) learns the importance of each modality with a softmax normalization. Both three-dimensional outputs are combined by a dot product, which are further reduced into a single dimension for making the final prediction in the last step. For a given clip, the model dynamically infers each modality's importance for making a prediction, which also provides interpretability for understanding the detection model's computational procedure to some extent.

Formally, the MINT model is defined as follows:

$$\begin{aligned} \mathbf{h}_{\text{imp}} &= [W_1 \bar{\mathbf{h}}_{\text{freq}}; W_2 \bar{\mathbf{h}}_{\text{chat}}; W_3 \bar{\mathbf{h}}_{\text{video}}], \\ \mathbf{h}_{\text{pred}} &= [W_4 \bar{\mathbf{h}}_{\text{freq}}; W_5 \bar{\mathbf{h}}_{\text{chat}}; W_6 \bar{\mathbf{h}}_{\text{video}}], \\ \mathbf{R} &= \text{Softmax}(\mathbf{h}_{\text{imp}}) \odot \mathbf{h}_{\text{pred}}, \\ \hat{y} &= \text{argmax}(W_{\text{out}} \mathbf{R}), \quad W_i \in \mathcal{R}^{h \times 1} (i = 1, 2, \dots, 6), W_{\text{out}} \in \mathcal{R}^{2 \times 3}, \end{aligned} \quad (4)$$

where  $\bar{\mathbf{h}}_{\text{freq}}$ ,  $\bar{\mathbf{h}}_{\text{chat}}$ , and  $\bar{\mathbf{h}}_{\text{video}}$  are hidden vectors after the feature representation step. Softmax represents the conventional softmax normalization.  $\mathbf{h}_{\text{imp}}$  and  $\mathbf{h}_{\text{pred}}$  are 3-dimensional outputs corresponding to the importance of each feature and prediction score, respectively.  $R$  is the combined outputs incorporating the importance of each data modality, and  $\hat{y}$  indicates the classification output. When training MINT, all parameters, including the networks for feature representation, are jointly trained to minimize a training objective. The implementation code is available in the Additional file 2.

### 5.2.5 Feature-based models

As a baseline, we present three feature-based models that separately learn each data modality. We employ logistic regression using a different set of features. For the chat frequency model, we use the averaged  $\mathbf{v}$  as input features. Word and emote embedding vectors are averaged for the chat content model used for input features. For the visual content model, we extract features for each frame using the Visual Geometry Group network (VGGnet) [46], which has been widely used in the recent literature as in [47, 48], with 16 layers that were pre-trained on ImageNet and averaged the features for logistic regression input.

## 5.3 Evaluation of classification performance

We evaluate the classification performance of the proposed models. Since the clip view count exhibits a long-tail distribution, we conduct a balanced sampling of clips. Using random undersampling, we have a balanced set of 97,649 popular clips and 97,165 ordinary clips.<sup>5</sup>

Table 5 shows the descriptive statistics of the balanced dataset. The dataset is split into train, dev, and test set with the ratio of 8:1:1. For the deep learning model training, we use a batch size of 32 and a maximum epoch of 10. Each model is trained to minimize the loss function by using a stochastic gradient optimizer with a learning rate of 0.01. The deep neural networks are trained to minimize the parameters' cross-entropy function and one-hot label vectors.<sup>6</sup>

<sup>5</sup>Some popular clips were deleted during the data collection process, and thus we could not extract the visual features from them. We excluded those clips that are not available for visual features.

<sup>6</sup>We will release the code and implementation details with the final version of this paper.

**Table 5** Descriptive statistics of balanced dataset ( $N_{\text{video}} = 10,068$ ,  $N_{\text{channel}} = 201$ )

Type	Mean	Median	Min	Max
Number of clip per video	19.34	12	1	244
Number of clip per channel	969.22	628	8	6442

**Table 6** Evaluation result for each model

Type	Model	Accuracy			AUC		
		Train	Dev	Test	Train	Dev	Test
Feature-based	Chat frequency	0.6480	0.6555	0.6545	0.6963	0.7165	0.7159
	Chat content	0.6857	0.6820	0.6841	0.7477	0.7437	0.7449
	Video content	0.5653	0.6110	0.5478	0.6117	0.6114	0.6159
Deep learning	Chat frequency	0.8099	0.8089	0.8125	0.8720	0.8716	0.8748
	Chat content	0.7271	0.7057	0.7013	0.7969	0.7710	0.7699
	Video content	0.8210	0.8097	0.8140	<b>0.9094</b>	0.8879	0.8901
Multi-modal	MINT	<b>0.8328</b>	<b>0.8317</b>	<b>0.8261</b>	0.9054	<b>0.9074</b>	<b>0.9006</b>
	(w/o importance learning)	0.8237	0.8256	0.8173	0.9018	0.8964	0.8911

Table 6 shows the performance measured on train, validation, and test set in terms of accuracy and area under the receiver operating characteristic (ROC) curve (AUC). Accuracy measures the fraction of true prediction compared to the test set size based on the prediction threshold of 0.5. AUC provides a balanced measure of classification performance aggregated over all possible classification thresholds, resulting in a more robust score against the test set distribution. We consider the popular view count group as the positive label for measuring AUC. Additional evaluation results on precision, recall, and F1 score are available in the Additional file 1.

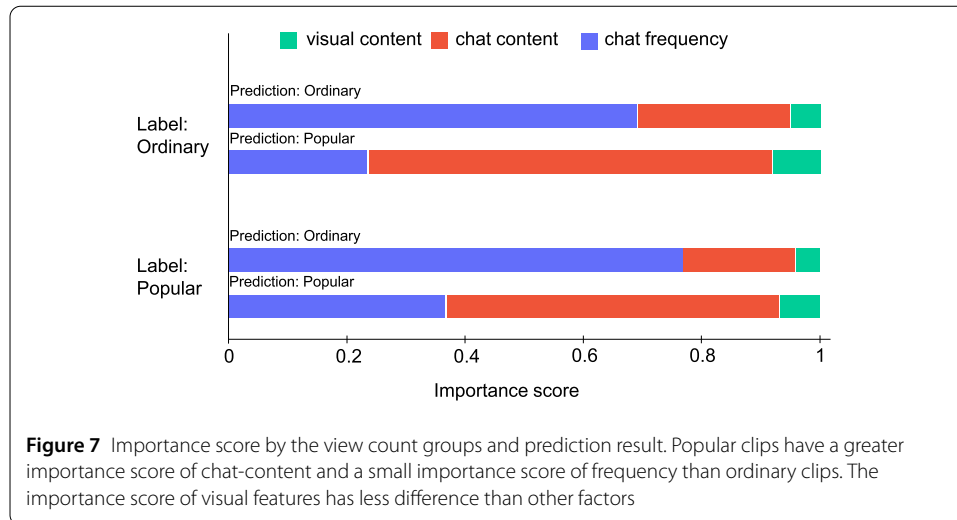
From the evaluation, we make three observations. First, for each data modality (e.g., chat frequency), the deep learning-based model achieves a better performance. For example, compared to the feature-based model, the deep learning-based video content model's performance gain is 0.2662 and 0.2742 for accuracy and AUC on the test set, respectively.

Second, among the deep learning-based models using each feature set, the video content model achieves the best result in terms of both metrics (0.814 and 0.8901); the chat frequency model achieves a similar performance (0.8125 and 0.8748). This observation implies that video content provides the most important cues for detecting popular moments, but the audience chat conversation also provides the equivalent information level.

Third, when the three different cues are combined, the MINT model achieves the best classification performance. Ablating the importance of MINT's learning mechanism reduces its performance to a similar video content-based model (0.8173 and 0.8911). The results suggest that importance learning is key to the MINT's performance.

We further test the interpretability of importance learning in MINT.  $h_{\text{pred}}$  in Eq. (4) contributes each data modality for combining the three different cues. From inference on each instance of the test set, we extract the three-dimensional vector  $h_{\text{pred}}$ , which we call *importance score* for the rest of this paper. Figure 7 presents the score distribution across different combinations of the view count group and model prediction. The score distribution varies for each predicted instance, and thus we use each class's mean value in the figure. The results show that, on average, chat frequency is the most important feature for predicting ordinary moments. On the other hand, chat content becomes the most important for predicting popular moments.





Our dataset covers diverse game domains. Since the overall popularity varies across different games, a data-driven model can capture a game's patterns rather than focus on the difference between popular and ordinary moments. We evaluate classification performance by using a game-specific dataset of three popular games: Fortnite (35,989), Grand Theft Auto V (34,507), and League of Legends (17,113). Using the same split as the main experiment, the MINT model achieves the test accuracy of 0.81, 0.84, and 0.88 for each game-specific data, suggesting its robustness.

## 6 Epic moment detection and user study

We now present an algorithm for the extraction of epic moments, which is defined as the collection of most enjoyable moments. We also conduct a user study for evaluating its performance in the wild, by comparing the algorithmic clip recommendations with the expert human-generated suggestions.

### 6.1 Algorithm

Algorithm 1 describes the procedure of finding epic moments from user-generated live streams using a classification model trained for discriminating popular and ordinary clips. This algorithm recommends video segments that will likely incur high view counts by learning from the audience's reactions. The algorithm scans the entire video and infers the likelihood of receiving high view counts for a 30-second window by sliding it every 10 seconds. Once the scan is complete, Algorithm 2 selects the top- $k$  moments in terms of the predicted score and merges them for generating an epic moment for a given video.

### 6.2 User survey setups

We evaluate the algorithmic suggestion such that the test dataset does not overlap with the live streaming content used in the training dataset. We randomly sampled five new live streaming content for each channel in our dataset and verified which content has the matching expert recommendations on social media. One such example is highlight videos posted to YouTube, which refer to an editor-generated high-quality summary of the long live streams and can be considered a gold standard of epic moments. The highlight content

---

**Algorithm 1:** Epic moment detection

---

**Input:** video frames  $V$  (list), chat  $C$  (list), video length  $L$  (second), classification model  $M$

**Output:** prediction scores for every 30-second long window

```

scores = <>;
for t = 0 to  $\lceil \frac{L}{10} \rceil$  do
    start = t × 10;
    end = min(start + 30, L);
    target_frames = V[start : end];
    target_chat = C[start : end];
    scores.append(M(target_frames, target_chat));
end
return scores;
```

---



---

**Algorithm 2:** Top k epic moment selection

---

**Input:** model prediction scores  $scores$  (list), number of target moments  $K$  (integer), video length  $L$  (second)

**Output:** top  $K$  epic moments for given video

```

index = argsort(scores, descending = True) // return sorted indices in
the descending order
for t = 0 to K - 1 do
    start = index[t] × 10;
    end = min(start + 30, L);
    top_k_moments.append((start, end));
end
return top_k_moments;
```

---

is mostly linked to the Twitch channel.<sup>7</sup> They can last several minutes to over ten minutes. We downloaded the videos and analyzed every image frame to identify which segment of the original live streaming content had been edited and re-posted on YouTube. We call these segments *expert-selected*. In total, we collected 1483 expert-selected epic moments from 34 highlight videos on YouTube.

For the newly collected videos and chats, we applied Algorithm 1 to identify epic moments suggested by MINT. MINT is chosen as it achieves the highest performance in evaluation. Using Algorithm 1's outputs, we adopted the top- $k$  suggestions for the *machine-selected* epic moments. From the pool of expert- and machine-selected clip candidates, we randomly chose 60 content pairs and used them for the user study.

Using the paired dataset of *machine-selected* and *expert-selected* suggestions, we conducted a crowdsourced experiment and asked about their impressions on the content. We asked the following questions: (1) Which clip do you prefer to watch? and (2) What kind

---


<sup>7</sup>For example, a Twitch video (<https://www.twitch.tv/videos/307453506>) has a link to the highlight video shared on YouTube (e.g., <https://www.youtube.com/watch?v=Zviuz-5qTUE>)

**Table 7** Survey results ( $N = 60$ )

Approach	Q1 (preference)	Q2 (funny)	Q3 (surprising)	Q4 (sad)	Q5 (exciting)	Q6 (nervous)
Expert-selected	30	23	26	26	35	33
Machine-selected	23	32	27	20	24	21
Neither	7	5	7	14	1	6

of emotion does the clip contain? We hired the survey respondents in Prolific and asked the participants to watch a pair of 30-second clips and respond to the survey questions.

Below is the detailed questions we asked in the experiments:

(Pre-test) What is the meaning of this emote ?

- (Q1) To what extent do you prefer the first clip to the next one?
- (Q2) Which clip do you think contains funny emotion more?
- (Q3) Which clip do you think contains surprising emotion more?
- (Q4) Which clip do you think contains sad emotion more?
- (Q5) Which clip do you think contains exciting emotion more?
- (Q6) Which clip do you think contains nervous emotion more?
- (Q7) Do you remember the channel name of each clip?

The first question was not related to the study directly but was a condition question to ensure that the survey participant is aware of the Twitch platform. The Kappa emote is used widely within Twitch to convey a meaning of ‘sarcasm’ and is known as one of the most community-specific norms. In case a participant did not guess Kappa’s meaning correctly, we discarded the response from the analysis. We ensured that participants spend enough time watching the video by checking the video playback time for all qualified survey participants. Once a pair of clips (i.e., machine-selected and expert-selected choices from the same live stream) was watched, the participant was then asked survey questions. We asked six questions related to which clip is preferred (Q1) and which clip covers a given emotion (Q2–Q6). To ensure that participants did not respond randomly, we also added an attention-check question (Q7) that asked participants to recollect the live streaming channel’s name. We ran this survey until a minimum of five respondents (who passed the attention check question) were obtained for each of the 60 clip pairs.

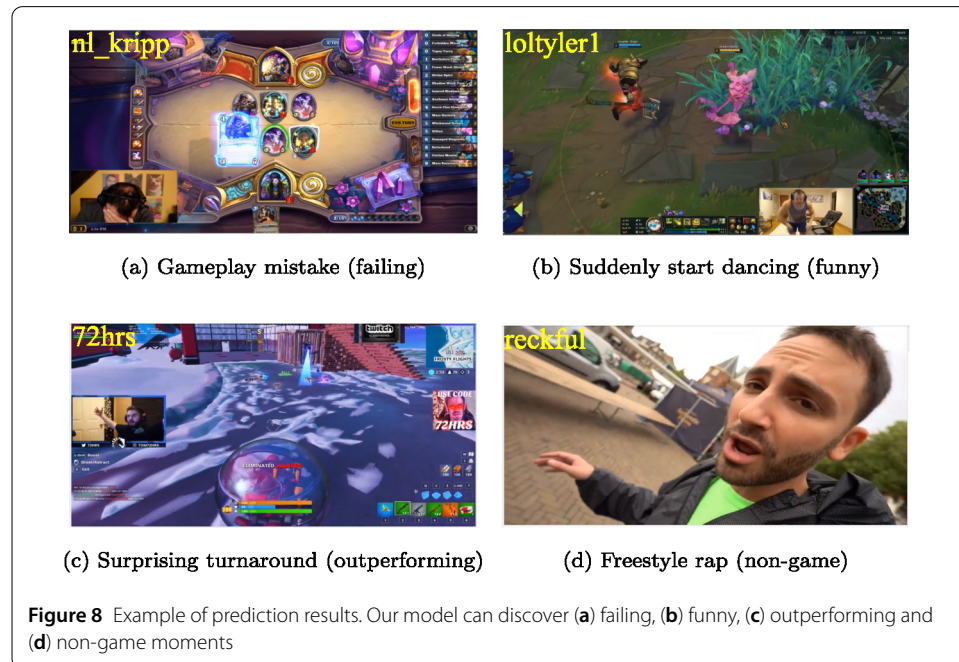
### 6.3 How well does our approach identify epic moments?

Table 7 presents the survey results. Before the analysis, we aggregated the survey responses by taking the majority voting. Related to Q1 on clip preference, the expert-selected suggestions were preferred on average (61%). Yet, we interpret that, notably, machine-selected suggestions are comparable to expert-selected suggestions; 23 machine-selected clips out of 60 pairs had a winning score. Next, concerning Q2–Q6 on the presence of emotion, the expert-selected and machine-selected suggestions were nearly equally perceived as containing emotions of any kind.

Having found that machine suggestions are comparable to expert suggestions, we examined how different they are in the covered emotion. Table 8 shows the degree of audience response (according to the emote clusters) each clip type was rated to contain. The analysis finds that 73% (or 44 out of 60) of the expert-selected clips contain at least one of the emotes from the ‘funny’ emotion cluster, whereas only 53% (or 32 out of 60) of the machine-selected clips did so. When it comes to the surprising emotes, machine-selected clips were better at capturing such emotion. There was not much difference in the cov-

**Table 8** Number of video moments where each emote cluster appears in the audience chat. Note that an audience chat can include none or multiple emote clusters

Approach	Funny	Surprising	Nervous	Sad	Exciting
Expert-selected	44	10	8	4	10
Machine-selected	32	21	10	3	8



erage between the expert-selected and machine-selected suggestions for the other three emotions (i.e., nervous, sad, and exciting).

This finding shows that the collective audience responses and the meta-information about the live content channel can be jointly used to identify epic moments of diverse contexts from lengthy live content. It is even equivalent to expert-selected suggestions, and this implies a potential to develop a more accurate model that mines seemingly mundane content of everyday users to identify epic moments. Since manual selection is costly and non-scalable, our findings and the potential for an automated epic moment extraction will help the broadcasters, channel fans, and service providers.

Figure 8 displays several examples of machine-selected clips and one example description of the scene given by survey respondents. Figure 8(a) captures a moment of loss for the broadcaster due to a gameplay mistake. Several respondents described it as *“This is a funny failing moment, and it is more fun to lose like this!”*, *“The <broadcaster> tries to recover from the mistake but ends up damaging himself.”* Next, Fig. 8(b) was recognized as a funny moment by many participants; the broadcaster (shown inside the small screen) suddenly started dancing with the game character. One of the respondents’ described it as *“His character starts dancing, and upbeat techno music is playing, and the streamer gets up and starts dancing too.”*

Figure 8(c) is a scene where the broadcaster is suddenly attacked inside the game, but he successfully defended against the attack. The scene was described as *“That was a surprising turnaround. I also like how the broadcaster gave a comedic reaction afterward.”*

Fig. 8(d) is a scene that is not relevant to the gameplay; in this clip, the broadcaster is doing freestyle rapping and engaging with the audience. These various moments of failing gameplay, turn around, and interaction were identified as epic moments by our algorithm. The expert-selected suggestions were more likely to contain victorious game moments in contrast. These examples illustrate that our model can select diverse moments ranging from specific game-related moments to funny moments.

## 7 Conclusion

Streaming services have become a significant media platform. The Twitch audience members interact with each other through chat and have developed community norms by using specific emotes to engage over streamed content. Users can also create ‘clips’ or short segments of live streaming content that are epic. These user-generated clips function as a measure of popularity.

This study collected clips of the content streamed on Twitch.tv and analyzed them together with other metadata from the original content. One important type of metadata we examined consisted of chat logs that accompany the live-streaming video. Based on the insight that audience reaction is a surrogate for estimating highlights in live streams, we suggest using clip features to capture the audience’s chat logs responses. The proposed clip features contain the structural patterns of audience reaction, semantic meanings, and sentiments. These clip features differ based on the clip popularity level. We introduced an emote embedding method and found the tendency of clip emotion types based on popularity. We also suggested a deep learning-based model with three domains(chat, frequency, and visual contents) to predict popular moments. To train the model, we utilized user-defined clips and the corresponding view counts. We applied the deep learning model to discover epic moments over a long stream, representing enjoyable moments. The user study comparing the machine-generated outcomes with human-expert selections imply that our approach can discover epic moments containing diverse emotions.

Predicting content popularity has been a focus of study in various domains from various aspects. Some previous studies have captured creator-edited materials such as videos on YouTube [6]. Other works have suggested choosing highlight moments from live streams using multiple clues from their interfaces, such as webcams, IRC chat rooms, or gameplay screens [47]. However, to the best of our knowledge, no prior study or its dataset has utilized user-edited moments such as clips captured from live streaming videos. Thus, this study contributes to understanding essential characteristics of clip datasets and capitalizing on audience sentiment to predict popular clips. We proposed a simple way to improve the emote embedding to capture the audience sentiment, resulting in seven clip types.

Our findings include intensive user interaction on popular moments such as high utterance frequency, short messages, and incoherent chat content. These characteristics have been observed across various social platforms [14, 39]. Twitch users express their emotions during the epic moments, generating a spike in chat utterance [39] and constructing crowdspeak [14]. Conformity effect may explain this phenomenon, which matches the attitude or behaviors to the group norms in that previously silent users suddenly start to express their feelings when others do. This effect can also be considered as emotional contagion. As a previous study pointed out [17], a scene or event with certain emotion makes the viewers feel the same way, resulting in similar expressions in chat rooms. Investigating the audience behavior in chat from the perspectives of conformity or emotional contagion will be a fascinating future research direction.

The proposed model's performance can be improved from multiple perspectives, including filtering the inputs or applying more advanced deep learning architectures. For instance, we could separate a gameplay screen from a broadcaster webcam. It is also possible to introduce sound features to help predict epic moments. Additionally, introducing more advanced natural language processing approaches for processing the chat logs can be advantageous. It is possible to use advanced general-purpose language models like bidirectional encoder representations from transformers (BERT [49]) to process the audience reactions. Advanced sentiment analysis techniques may be employed for identifying sentiment toward a streamer or other users [50]. Among the user features, social relationship metrics such as channels followed or friend relationships may be used.

This study is limited by its data. We targeted popular English-speaking male broadcasters due to the small female proportions in the top-200 streamers, introducing cultural bias. We believe it is an important future direction to understand how epic moments' characteristics become different for female streamers. Therefore, to facilitate future analyses, we report the gender difference in chat characteristics in the Additional file 1 and also release the top 1000 word embedding vectors trained on the studied male-only dataset as well as the dataset including both genders. In addition, the user study allows us to discuss the differences between the existing audience and those who are new to the channel in terms of preference or sense of emotions. It will be interesting to understand how Twitch users react to the machine-generated outputs according to their loyalty and familiarity.

We believe our approach can be applied to Twitch or similar live streaming services. It helps streamers improve their stream quality by reviewing the epic moments suggested by the machine learning model, enabling them to know which moments are epic and therefore will gain popularity when re-created as a short clip. The interpretability of our model gives additional insights on potential cues for epic moments, and an understanding of such interpretability will help streamers produce better quality streaming. Yet, since our method mainly utilizes audience reactions, it does not apply to less popular channels (i.e., owning fewer viewers) due to lack of audience reactions and clips. Therefore, to facilitate future analyses, we report the gender difference in chat characteristics in the Additional file 1 and also release the top 1000 word embedding vectors in the Additional file 2, which are trained on the studied male-only dataset as well as the dataset including both genders.

## Supplementary information

**Supplementary information** accompanies this paper at <https://doi.org/10.1140/epjds/s13688-021-00295-6>.

**Additional file 1.** Supplementary information (PDF 329 kB)

**Additional file 2.** Supplementary material (codes and embeddings) (ZIP 1.0 MB)

## Acknowledgements

The authors thank to Karandeep Singh, Sungwon Han, Sundong Kim, and Sungwon Park for their helpful comments.

## Funding

H. Song and M. Cha were supported by the Institute for Basic Science (IBS-R029-C2) and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. NRF-2017R1E1A1A01076400). K. Park was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. NRF-2021R1F1A1062691).

## Abbreviations

SVM, support vector machine; CNN, convolution neural network; LSTM, long short-term memory; EMTM, emoticon-aware topic model; STC, Sparse Topical Coding; API, application programming interface; CCDF, complementary cumulative

distribution function; ANOVA, analysis of variance; Turkey HSD, Turkey honestly significant difference; IRC, Internet relay chat; NLTK, Natural language toolkit; CBOW, continuous bag-of-words; DBSCAN, density-based spatial clustering of applications with noise; BTTV, BetterTTV; FFZ, FrankerFaceZ; t-SNE, t-distributed stochastic neighbor embedding; RNN, recurrent neural network; EM-DeEp, ensembling multi-modal detection of epic moment; VGG, visual geometry group; BERT, bidirectional encoder representations from transformers.

#### Availability of data and materials

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

#### Competing interests

The authors declare that they have no competing interests.

#### Consent for publication

This paper meets the guidelines from Twitch's terms of service that there are no copyright or privacy infringement issues regarding the use of the images in the paper.

#### Authors' contributions

KP and MC conceptualized this research. HS collected and analyzed the data, and KP developed the neural network model. All authors contributed to the manuscript's writing and editing. All authors read and approved the final manuscript.

#### Author details

<sup>1</sup>School of Computing, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea. <sup>2</sup>Data Science Group, Institute of Basic Science, Daejeon, Republic of Korea. <sup>3</sup>School of AI Convergence, Soongsil University, Seoul, Republic of Korea.

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 3 December 2020 Accepted: 12 July 2021 Published online: 18 August 2021

#### References

1. Mansoor I (2021) Twitch revenue and usage statistics. <https://www.businessofapps.com/data/twitch-statistics/> (Accessed July 24, 2021)
2. Rochan M, Reddy MKK, Ye L, Wang Y (2020) Adaptive video highlight detection by learning from user history. In: European conference on computer vision. Springer, Berlin, pp 261–278
3. Drankou P, Zikov S (2020) Chat-based approach applied to automatic livestream highlight generation. In: International conference on intelligent decision technologies. Springer, Berlin, pp 437–447
4. Xiong B, Kalantidis Y, Ghadyaram D, Grauman K (2019) Less is more: learning highlight detection from video duration. In: Proc. of the IEEE conference on computer vision and pattern recognition, pp 1258–1267
5. Fu C-Y, Lee J, Bansal M, Berg A (2017) Video highlight prediction using audience chat reactions. In: Proc. of the conference on empirical methods in natural language processing, pp 972–978
6. Yao T, Mei T, Rui Y (2016) Highlight detection with pairwise deep ranking for first-person video summarization. In: Proc. of the IEEE conference on computer vision and pattern recognition, pp 982–990
7. Babaguchi N, Kawai Y, Ogura T, Kitahashi T (2004) Personalized abstraction of broadcasted American football video by highlight selection. *IEEE Trans Multimed* 6(4):575–586
8. Jiao Y, Li Z, Huang S, Yang X, Liu B, Zhang T (2018) Three-dimensional attention-based deep ranking model for video highlight detection. *IEEE Trans Multimed* 20(10):2693–2705
9. Kaytoue M, Silva A, Cerf L, Meira W Jr, Raissi C (2012) Watch me playing, I am a professional: a first study on video game live streaming. In: Proc. of the international conference on world wide web, pp 1181–1188
10. Scheibe K (2018) The impact of gamification in social live streaming services. In: International conference on social computing and social media. Springer, Berlin, pp 99–113
11. Wulf T, Schneider FM, Beckert S (2020) Watching players: an exploration of media enjoyment on twitch. *Games Cult* 15(3):328–346. <https://doi.org/10.1177/1555412018788161>
12. Hamilton WA, Garretson O, Kerne A (2014) Streaming on twitch: fostering participatory communities of play within live mixed media. In: Proc. of the ACM conference on human factors in computing systems, pp 1315–1324
13. Sheng JT, Kairam SR (2020) From virtual strangers to irl friends: relationship development in livestreaming communities on twitch. In: Proc. of the ACM conference on computer-supported cooperative work and social computing 4, pp 1–34
14. Ford C, Gardner D, Horgan LE, Liu C, Tsaasan AM, Nardi B, Rickman J (2017) Chat speed op pogchamp: practices of coherence in massive twitch chat. In: Extended abstracts on human factors in computing systems, pp 858–871
15. Nematzadeh A, Ciampaglia GL, Ahn Y-Y, Flammini A (2019) Information overload in group communication: from conversation to cacophony in the twitch chat. *R Soc Open Sci* 6(10):191412
16. Musabirov I, Bulygin D, Okopny P, Konstantinova K (2018) Between an arena and a sports bar: online chats of esports spectators. arXiv preprint. [arXiv:1801.02862](https://arxiv.org/abs/1801.02862)
17. Guo J, Fussell SR (2020) A preliminary study of emotional contagion in live streaming. In: Proc. of the ACM conference on computer supported cooperative work and social computing, pp 263–268
18. Seering J, Kraut R, Dabbish L (2017) Shaping pro and anti-social behavior on twitch through moderation and example-setting. In: Proc. of the ACM conference on computer supported cooperative work and social computing, pp 111–125

19. Flores-Saviaga C, Hammer J, Flores JP, Seering J, Reeves S, Savage S (2019) Audience and streamer participation at scale on twitch. In: Proc. of the ACM conference on hypertext and social media, pp 277–278
20. Nakandala SC, Ciampaglia GL, Su NM, Ahn Y-Y (2017) Gendered conversation in a social game-streaming platform. In: Proc. of the international AAAI conference on web and social, Media
21. Ruvalcaba O, Shulze J, Kim A, Berzenski SR, Otten MP (2018) Women's experiences in esports: gendered differences in peer and spectator feedback during competitive video game play. *J Sport Soc Iss* 42(4):295–311
22. Eisner B, Rocktäschel T, Augenstein I, Bošnjak M, Riedel S (2016) emoji2vec: learning emoji representations from their description. arXiv preprint. [arXiv:1609.08359](https://arxiv.org/abs/1609.08359)
23. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint. [arXiv:1301.3781](https://arxiv.org/abs/1301.3781)
24. Kobs K, Zehe A, Bernstetter A, Chibane J, Pfister J, Tritscher J, Hotho A (2020) Emote-controlled: obtaining implicit viewer feedback through emote-based sentiment analysis on comments of popular twitch. tv channels. *ACM Trans Soc Comput* 3(2):1–34
25. Gilbert C, Hutto E (2014) Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: Proc. of the AAAI international conference on weblogs and social, Media
26. Ha J-W, Kim A, Kim C, Park J, Kim S (2017) Automatic music highlight extraction using convolutional recurrent attention networks. arXiv preprint. [arXiv:1712.05901](https://arxiv.org/abs/1712.05901)
27. Hsieh L-C, Lee C-W, Chiu T-H, Hsu W (2012) Live semantic sport highlight detection based on analyzing tweets of Twitter. In: Proc. of the IEEE international conference on multimedia and expo, pp 949–954
28. Zhang C, Liu J, Ma M, Sun L, Li B (2017) Seeker: topic-aware viewing pattern prediction in crowdsourced interactive live streaming. In: Proc. of the ACM workshop on network and operating systems support for digital audio and video, pp 25–30
29. Chu W-T, Chou Y-C (2017) On broadcasted game video analysis: event detection, highlight detection, and highlight forecast. *Multimed Tools Appl* 76(7):9735–9758
30. Sun M, Farhadi A, Seitz S (2014) Ranking domain-specific highlights by analyzing edited videos. In: Proc. of the European conference on computer vision, pp 787–802
31. Makantasis K, Liapis A, Yannakakis GN (2019) From pixels to affect: a study on games and player experience. In: Proc. of the IEEE international conference on affective computing and intelligent interaction, pp 1–7
32. Melhart D, Gravina D, Yannakakis GN (2020) Moment-to-moment engagement prediction through the eyes of the observer: pubg streaming on twitch. In: Proc. of the international conference on the foundations of digital games, pp 1–10
33. TwitchMetric. <https://www.twitchmetrics.net/> (Accessed Jul 24, 2021)
34. Twitchstats. <https://www.twitchstats.net/> (Accessed Jul 24, 2021)
35. Musabirov I, Bulygin D, Okopny P, Konstantinova K (2018) Event-driven spectators' communication in massive esports online chats. In: Extended abstracts of the ACM CHI conference on human factors in computing systems, p 564
36. Wohn DY, Jough P, Eskander P, Siri JS, Shimobayashi M, Desai P (2019) Understanding digital patronage: why do people subscribe to streamers on twitch?. In: Proc. of the ACM annual symposium on computer-human interaction in play, pp 99–110
37. Kim J, Bae K, Park E, del Pobal AP (2019) Who will subscribe to my streaming channel? The case of twitch. In: Proc. of the ACM conference on computer supported cooperative work and social computing, pp 247–251
38. Kim J, Park K, Song H, Park JY, Cha M (2020) Learning how spectator reactions affect popularity on twitch. In: Proc. of the IEEE international conference on big data and smart computing, pp 147–154
39. Kairam S, Morris M, Teevan J, Liebling D, Dumais S (2013) Towards supporting search over trending events with social media. In: Proceedings of the international AAAI conference on web and social media, vol 7
40. Olejniczak J (2015) A linguistic study of language variety used on twitch. tv: descriptive and corpus-based approaches. In: Redefining community in intercultural context, pp 329–344
41. Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E (2015) Deep learning applications and challenges in big data analytics. *J Big Data* 2(1):1
42. Yoon S, Park K, Shin J, Lim H, Won S, Cha M, Jung K (2019) Detecting incongruity between news headline and body text via a deep hierarchical encoder. In: Proc. of the AAAI conference on artificial intelligence, vol 33, pp 791–800
43. Ringer C, Nicolaou MA (2018) Deep unsupervised multi-view detection of video game stream highlights. In: Proc. of the international conference on the foundations of digital games, pp 1–6
44. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proc. of the IEEE conference on computer vision and pattern recognition, pp 770–778
45. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: Proc. of the IEEE conference on computer vision and pattern recognition, pp 248–255
46. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
47. Ringer C, Walker JA, Nicolaou MA (2019) Multimodal joint emotion and game context recognition in league of legends livestreams. In: Proc. of the IEEE conference on games, pp 1–8
48. Roohi S, Mekler ED, Tavast M, Blomqvist T, Hämäläinen P (2019) Recognizing emotional expression in game streams. In: Proceedings of the annual symposium on computer-human interaction in play, pp 301–311
49. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint. [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
50. Park K, Pan Z, Joo J, (2021) Who Blames or Endorses Whom? Entity-to-Entity Directed Sentiment Extraction in News Text. [arXiv:2106.01033](https://arxiv.org/abs/2106.01033)