Comparing Learning Methodologies for Self-Supervised Audio-Visual Representation Learning

HACENE TERBOUCHE1,2, LIAM SCHONEVELD1, OISIN BENSON1, AND ALICE OTHMANI2

1Powder AI Research, 75001 Paris, France
2LISI, UPEC, Université Paris-Est, 94400 Vitry-sur-Seine, France

Corresponding author: Alice Othmani (alice.othmani@u-pec.fr)

This work was supported by Powder, a Deep Tech Startup.

ABSTRACT In recent years, the machine learning community has devoted an increasing attention to self-supervised learning. The performance gap between supervised and self-supervised has become increasingly narrow in many computer vision applications. In this paper, a new self-supervised approach is proposed for learning audio-visual representations from large databases of unlabeled videos. Our approach learns its representations by a combination of two tasks: unimodal and cross-modal. It uses a future prediction task, and learns to align its visual representations with its corresponding audio representations. To implement these tasks, three methodologies are assessed: contrastive learning, prototypical constrasting and redundancy reduction. The proposed approach is evaluated on a new publicly available dataset of videos captured from video game gameplay footage, called Videogame DB. On most downstream tasks, our method significantly outperforms baselines, demonstrating the real benefits of self-supervised learning in a real-world application.

INDEX TERMS Self-supervised learning, audiovisual correspondence, cross-modal video representation learning, future prediction, learning methodologies.

I. INTRODUCTION

Over the past decade, deep neural networks have shown great capability in many practical applications. Due to their ability to automatically learn features from raw data, state of the art results have been achieved in a variety of problems. These include image recognition, object detection and semantic segmentation [1]. Hence, it reduces the reliance on designing hand-crafted components which, in most cases, requires high domain knowledge skills. In computer vision, models pretrained on large-scale datasets such as ImageNet are often used as feature extractors and then fine-tuned for different target visual tasks. Fine-tuning those models might help them to converge faster and reduce the over-fitting problem, especially when the target dataset size is small. The reason is that low-level features are already learnt, and are generally common across a wide range of applications.

The success of deep convolutional neural networks (CNNs) hinges upon sophisticated architectures and the amount of training data. CNNs are usually trained in a fully supervised manner, where each data example has its corresponding label [1]. However data collection and annotation at a large scale is costly and time-consuming. Moreover, establishing large labeled video datasets is more challenging due to the time dimension. To mitigate against this many self-supervised learning have been proposed in order to learn visual representations from large unlabeled datasets without any human annotation. At a high level self-supervised approaches are trained to solve one or more pretext tasks. Pretext tasks are pre-designed tasks for models to solve in order to learn useful features by optimizing some proxy objective functions. They are generally not useful by themselves and they are used to extract semantic representations [2]. For instance the model might be shown data with some missing parts in the spatial or in the temporal dimension and be required to reconstruct the missing parts.

Unlabeled videos are widely available from different social media and streaming platforms on a very large scale. Hence, it is desirable to develop deep learning models to understand this content usefully. This will allow for a large range of applications with little to no annotated data including video understanding, recommendation systems and video...
indexing. Another practical application, video games highlighting, which aims to detect the specific events that are more attractive to the audience such as attaining a goal or a kill. Due to the popularity of game online and live streaming videos, it is becoming desirable to develop a machine learning model for automatic highlight detection. Hence, the role of self-supervised learning is to design the suitable proxy tasks to learn powerful representations. Recent visual representation learning methods are based on instance contrastive learning wherein an augmentation module generates multiple, semantically-equivalent views of the data [3]. Then the model is required to align its learned representations of these different views. Thus, those methods effectively rely on data augmentation to learn efficient features.

In general, humans experience the world through multi-simultaneous signal sources. For instance, lips moving is often referred to human talking and the sound of waves induces a scene of a beach. In the gaming context there is often a special sound effect that coincides with the visual content. This correlation between the visual and audio components of a video might be a useful signal of knowledge. In this work we leverage this intrinsic property of videos by utilizing a self-supervised task that attempts to compare the visual representations against multiple audio representations (and vice versa).

The sequential structure of videos offers a temporal information and spatio-temporal dynamics representations to learn. For instance the model can be required to predict the future to learn those representations. In fact a limited number of sequence can be fed to the learning model to train to predict the next element of the sequence. The shared features in the frames of the sequences are extracted. Moreover the low-level features of each sequence item are ignored in order to be more robust in predicting the future. In this work a simple future prediction learning system that can be applied to any modality having the temporal dimension is proposed. The proposed approach is evaluated under two modalities: RGB frames and audio data.

Despite the recent popularity of self-supervised learning approaches to computer vision in machine learning research, to our knowledge, the extent to which such approaches have been applied to more ‘real world’ problems is still somewhat limited. Moreover research in this sub-field has primarily focused on academic benchmarks, namely ImageNet. While valuable as a common performance metric, they provide little insight into the overall generality of these methods which theoretically should be applicable to a much broader range of tasks.

Motivated by the facts above, a new self-supervised approach is proposed to learn both visual and audio semantic representations by simply looking at and listening to a large number of unlabeled videos. In fact our goal is to lower the labeling cost for downstream tasks of interest. First the proposed approach leverages the multi-modal structure of videos by learning how to align the audio and the visual representations. Second the approach exploits the temporal dynamics of video data by learning the relevant information and representations for future prediction. We apply this approach to a new, partially-labeled real-world dataset of video game gameplay footage and release this dataset. In theory the proposed approach can be applied to any video-with-audio data. The main contributions of this paper can be summarized as follows:

- A new deep neural network-based framework for audio-visual feature learning is proposed. This network learns two feature extractors – one for audio and one for RGB frames – that can later be separately or jointly applied to any downstream task (audio, visual, or audio-visual) of interest
- A new dataset for video representation learning in the gaming context: Videogame DB.
- A simple and effective approach for applying a contrastive loss for future prediction and audio-visual correspondence, simultaneously
- Evaluating the performance of redundancy reduction as a loss to solve the audio-visual alignment task
- Introducing cross-stream prototypical contrasting as an alternative to instance contrasting.

The main contributions of this paper can be summarized as follows:

II. RELATED WORK

Self-supervised learning has long been a main driver in the advancement of natural language processing (NLP), including Word2Vec [4], GloVe [5], fastText [6], and, more recently, BERT [7] and GPT-3 [8]. However self-supervised approaches to computer vision have become increasingly successful. In this section we give an overview of these approaches starting with image applications and then continuing to video and finally to multimodal audio-visual applications.

A. SELF-SUPERVISED IMAGE REPRESENTATION LEARNING

Early works learn visual representations from unlabeled images mainly based on reconstruction methods. Among these methods, the auto-encoder [21] models learn to encode the image in a low dimensional vector which then is uncompressed into the raw input image. The encoder part of the auto-encoder is used as a feature extractor for further fine-tuning. This family of approaches also includes multi-layer auto-encoders, convolutional auto-encodes and restricted Boltzmann machines [22]. Current generation methods are largely based on generative adversarial networks (GAN) [23]. GANs are made up of a generator and a discriminator that have been trained in a two-player min-max game. Since the discriminator should distinguish between real and fake images and so it captures image semantics, it may be used as a pretrained model for other tasks.

Another line of representation learning methods defines an explicit proxy task called the pretext task to solve in order to learn visual features, such as image inpainting [24] or image colorization [25]. The spatial structure of the image can be a...
| Year | Ref | Modality | Pretext Tasks | Approach | Pretraining Dataset | Downstream Tasks | Downstream Dataset |
|------|-----|----------|--------------|----------|--------------------|------------------|-------------------|
| 2019 | CFN [9] | Image | Jigsaw puzzle | Learn a siamese-ennead network to solve jigsaw puzzle permutations of image’s patches | ImageNet | Image Classification | PASCAL VOC |
| 2019 | Deep Cluster [10] | Image | Clustering | Cluster data with k-means and use the assignments as a supervision to learn the model’s parameters | ImageNet | Image Classification | PASCAL VOC |
| 2019 | SimCLR [3] | Image | Instance Discrimination | Maximize the agreement between differently augmented views of the same data example via a contrastive loss in the latent space | ImageNet | Image Classification | ImageNet and 12 natural image dataset |
| 2021 | DINO [11] | Image | Self-distillation with no labels | Maximize the similarity between the teacher and the student networks, that is used as a supervision to the student model, while the teacher is updated with an exponential moving average | ImageNet | Image Classification | ImageNet GLDv2 | Copy detection |
| 2021 | SwAV [12] | Image | Online clustering with contrastive learning | Cluster the data using an online algorithm while simultaneously solving the swapped prediction problem by predicting the code of a view from the representation of another view | ImageNet | Image Classification | ImageNet VOC07+12 | COCO |
| 2021 | Barlow Twins [13] | Image | Redundancy reduction | Reduce the redundancy of the components of the feature vector and increase the agreement between the two augmentations using the cross-correlation matrix | ImageNet | Image Classification | ImageNet VOC07+12 | COCO |
| 2021 | CVRL [14] | Video | Instance Discrimination | Pull together two augmented clips from the same short video and push away clips from different videos in the embedding space | Kinetics-600 | Action classification | Kinetics-600 UCIF-101 HMDB-51 |
| 2020 | PRP [15] | Video | Discriminative & Generative Perception | Predict the playback rate of of sampled clips and reconstruct the slow-down video clips using a decoder network | UCF-101 | Action classification | UCF-101 | HMDB-51 |
| 2020 | O3N [16] | Video | video order verification | Identify the odd video clip (wrong temporal order) from a set of several video sequences using a multi branched network that share weights | UCF-101 HMDB-51 | Action classification | UCF-101 HMDB-51 |
| 2019 | XDC [17] | Audio-Video | Clustering | Leverage clustering in one modality (audio) as a supervisory signal for the other modality (video) and vice versa | Kinetics AudioSet IG-Kinetics IR-Random | Video action recognition | UCF-101 HMDB-51 EESC50 |
| 2021 | ViCC [18] | RGB-Flow | Cross-stream Prototypical Contrasting | Predict the assignments of one view (RGB) from the other view (optical flow) and vice versa. | UCF-101 | Video action recognition | UCF-101 HMDB-51 |
| 2021 | CoCLR [19] | RGB-Flow | Multi-instance contrastive learning | Mine more positive samples of one view from the other view for contrastive learning and use the co-training mechanism to train the two streams encoders | UCF-101 Kinetics-400 | Action classification | UCF-101 Kinetics-400 HMDB51 |
| 2021 | AVID-CMA [20] | RGB-Audio | Instance Discrimination | Discriminate the corresponding audio vector of the video clip and vice versa. Add cross-modal agreement for better positives and negatives | AudioSet Kinetics | Action recognition | UCF-101 HMDB-51 EESC DCASB |
rich context information to design proxy tasks for image feature learning. Images can be divided into multiple patches, allowing for tasks such as solving an image jigsaw puzzle [9], predicting the relative positions of two patches from the same image [26], or recognizing the order of the shuffled sequence of patches from the same image. Others tasks take benefits from the entire structure of the image such as recognizing the rotating angles of the whole image [27].

A number of researchers have also investigated the use of clustering methods as a proxy task [28], [28]. Clustering is a very common method in unsupervised machine learning to group sets of similar data points in the same cluster. For instance, Deep Cluster [28] is a self-supervised approach that iteratively clusters images using k-means algorithm to get the pseudo-labels and uses the subsequent assignments of the samples as a supervisory signal to optimize the parameters of the CNN model. Other methods train the model to recognize whether two images belong to the same cluster [29].

Instance discrimination is another strategy that trains an augmentation-invariant classifier by considering each sample in the dataset as though it were its own class [30]. For large-scale datasets, it becomes computationally demanding as the softmax computation needs to sum up over the total number of classes. To address this, Wu et al. [31] proposed a better formulation of the problem using non-parametric classifier and noise-contrastive estimation (NCE) [32].

Most of the recent pretext tasks are derived from this formulation of instance discrimination. They are based on the so-called Siamese networks which is composed of two identical copies of the same architecture. In [3], a new framework called SimCLR is proposed, that learns the representations of images by maximizing the agreement between two random augmentations of the same image using contrastive learning. Other works that are based on augmentation include SimSiam [33] and SwAV [12]. Another family of methods introduces the idea of a momentum encoder, wherein one of the copies in the Siamese network is updated using exponential moving average of the other such as MoCo [34], BYOL [35] and DINO [11].

B. SELF-SUPERVISED VIDEO REPRESENTATION LEARNING

Self-supervised learning has been also applied to videos due to the large availability of unlabeled videos on the web. When compared to images, videos also include additional time dimension, making it a spatio-temporal signal. As a result, it provides greater benefits for designing useful proxy tasks while simultaneously presenting various challenges for learning efficient visual representations. Some proxy tasks that have been already applied to images can be extended to videos such as video generation [36] or video colorization [37].

Various pretext tasks are proposed to exploit the natural temporal order of frames which can be used as supervision signal for self-supervised feature learning. These include temporal order verification [38], temporal order recognition [39] and playback speed detection [40]. Temporal order verification aims to verify whether a sequence of frames is in the correct/incorrect temporal order and thus solving a binary classification problem to distinguish between correct and shuffled order sequences. On the contrary, temporal order recognition task requires to predict the order of the input sequence of frames among a fixed set of combinations. In playback rate detection task, the aim of the model is to predict the speed of the frames among a predefined set of rates by artificially changing the video frame rate of the sequence input.

Due to the success of contrastive methods in image feature learning [3], a new method referred as Contrastive Video Representation Learning (CVRL) is proposed in [14] for video feature learning. Since the rich information of the SimCLR comes mainly from the augmentation module, the design of both temporal and spatial augmentations for videos is critical. CVRL provides a time consistent, spatial augmentation approach for imposing severe spatial augmentations on each frame while preserving temporal consistency between frames.

Many studies have investigated learning features by exploiting RGB frame sequences in videos, which allows for video future prediction tasks [41]. Video prediction task consists of predicting one or more future frames based on a limited number of past frames. Most of those approaches are built on an encoder-decoder architecture, where the encoder encodes the spatial and the temporal features of the input sequence into a fixed-length vector to be fed to the decoder to generate the future frames. Differently, Oord et al. propose a universal approach referred as contrastive predictive coding (CPC) to learn representation of any sequential-like data by predicting the future in the embedding space based on autoregressive models and contrastive loss [42].

C. CROSS-MODAL VIDEO REPRESENTATION LEARNING

Cross-modal learning methods focus on using several signals or views of the same visual content to learn the video features. It typically learns from the correlation of two or more stream modalities including RGB frames, optical flow, audio signal and text.

Besides the rich information that can be extracted from the individual visual frames, optical flow can be computed on the fly from sequence of images to describe the motion of objects between adjacent frames. Consequently, many works have been proposed to use the alignment between optical flow stream and RGB frames. This type of pretext tasks includes optical flow estimation [43] and RGB and optical flow correspondence verification [44]. In the same line of the SimCLR framework, Han et al. [19] have proposed a new self-supervised co-training method called CoCLR, in which contrastive learning with positive samples mining is used to align RGB and optical flow data. Toering et al. combine the co-training mechanism and the idea of the SwAV approach [12] to propose prototypical contrastive learning [18] as an alternative to instance-based contrastive learning.
Similarly, audio data can provide a useful information about the content of videos. It is then worthwhile to use the alignment between the two streams as a source of supervision for video self-supervised learning. This task is most called Audio-Visual Correspondence (AVC). AVC is a proxy task that attempts to capture the correlation between an image or a sequence of images (short clip) with the synchronised audio clip. The general idea of AVC is to have an encoder network for each modality and then fuse the two encodings to compute some loss function that penalizes unrelated (out-sync) clips while the two networks are learnt jointly. Many works have been proposed that formulate this task as a binary correspondence (AVC) task is explained in section III-A, then, learning useful features of the data. Applying then a projection MLP on each side separately \((g^v(\cdot), g^a(\cdot))\), to produce the vectors \(z^v \in \mathbb{R}^{d_z}\) and \(z^a \in \mathbb{R}^{d_z}\) respectively. They are computed as follows:

\[
\begin{align*}
z^v &= g^v(h^v) \\
z^a &= g^a(h^a)
\end{align*}
\]

As depicted in Figure 1, this task tries to match the two modalities’ representations. Multiple learning procedures are proposed in this work in order to solve this task and they are discussed in the next section.

III. OUR SELF-SUPERVISED LEARNING APPROACH

In this paper, a new approach for audio-visual self-supervised learning is proposed. In the following, Audio-Visual Correspondence (AVC) task is explained in section III-A, then, the second component of the framework, Future Prediction (FP), is detailed in section III-A. The combination of both tasks is then framed into a multi-task cross-modal system in section III-C.

A. AUDIO-VISUAL CORRESPONDENCE

Audio-Visual Correspondence task attempts to maximize the agreement between the the representation of a visual clip (RGB sequence) with that of an audio clip. AVC aims to capture the shared information between the two streams. Consider a short video clip with few seconds \(x\), the RGB sequence \(x^v\) and the audio clip \(x^a\) are then extracted from \(x\). For each stream, an encoder network is implemented, \(f^v(\cdot)\) as a visual encoder and \(f^a(\cdot)\) as an audio encoder. The inputs \((x^v, x^a)\) are processed through the two encoders in order to extract the context vector representations \(h^v \in \mathbb{R}^{d_1}\) and \(h^a \in \mathbb{R}^{d_1}\) respectively. \(h^v\) is the representation of the RGB sequence while \(h^a\) is that of the audio input. This can be written as follows:

\[
\begin{align*}
h^v &= f^v(x^v) \\
h^a &= f^a(x^a)
\end{align*}
\]

Following previous works such as [3], a Multi-Layer Perceptron (MLP) projection network for each modality is added to map the representations to the space of the proxy task. The projection heads aim to decouple solving the AVC task from learning useful features of the data. Applying then a projection MLP on each side separately \((g^v(\cdot), g^a(\cdot))\), to produce the vectors \(z^v \in \mathbb{R}^{d_z}\) and \(z^a \in \mathbb{R}^{d_z}\) respectively. They are computed as follows:

\[
\begin{align*}
z^v &= g^v(h^v) \\
z^a &= g^a(h^a)
\end{align*}
\]

B. FUTURE PREDICTION

To further extend our work, the AVC task is combined with another task: a new framework for predicting the future in both audio and visual spaces is proposed. Unlike previous works, where a generative model is used to predict the future frames of the sequence (in the input space) using a decoder \([41], [47]–[49]\), our approach performs the future prediction in in the embedding space (as per \([42]\)). This signifies that the feature vector of the next frame, rather than the frame itself, is predicted.

Figure 2 shows the architecture for the future prediction task for a single modality. Let \(x^m\) be a sequence of elements \([x^m_1, \ldots, x^m_{T_m}]\) from one stream \((m)\), where \(t\) is the time index...
Comparing Learning Methodologies

**FIGURE 2.** Future prediction architecture: this architecture is independent of the modality. The same setup can be applied to images, audio, text or optical flow. The sequence except the last element is encoded by the modality encoder \( f^m \) (to produce the context vector \( h_m^T \)), while the last element is fed to the CNN modality encoder \( f_m^{CNN} \). Two projection heads are applied on those representations to compute the loss in the latent space. The model maximizes the agreements between the two branches for encoding the sequence, and the other to embed the last frame of the sequence.

and \( T \) is the sequence length. First, a convolutional neural network \( f_{CNN}^m(\cdot) \) maps the input sequence into a sequence of representations \([h_m^1, \ldots, h_m^T]\), such as \( h_m^T = f_{CNN}^m(x_m^T) \). Then, a temporal model aggregates all the vectors \( h_{1:T}^m \) and outputs a context vector \( h_m^T \) called \( h_{true}^m \) with the predicted frame \( h_m^T \) in the latent space \( \mathbb{Z} \). Compared to \cite{Hjelm2018} who used only a linear projection on \( h_m^T \), an MLP projection head is added for each part and the prediction of the future is limited to one frame next.

\[
\begin{align*}
\hat{z}_{pred}^m &= s_{pred}(h_m^T) \\
\hat{z}_{true}^m &= s_{true}(h_{true}^m)
\end{align*}
\]

**C. JOINT AUDIO-VISUAL CORRESPONDENCE AND FUTURE PREDICTION**

A multi-modal, multi-task learning approach that combines audio-visual correspondence task and future prediction for both modalities is proposed as shown in Fig 3. This framework is hypothesized to take benefits from the two pretext tasks and thus leads to more effective representation of the data. As illustrated in Fig 3, this method consists of three components:

- The AVC task between the audio stream and the RGB stream. The only difference compared to Figure 1 is that the sequence except the last frame from each stream is fed to the model, instead of the whole sequence, allowing to use the last frame for the task of future prediction.
- The audio future prediction task same as in Figure 2 (replacing the superscript \( m \) by the superscript \( a \)).
- The visual future prediction task same as in Figure 2 (replacing the superscript \( m \) by the superscript \( v \)).

**IV. LEARNING METHODOLOGIES**

In order to solve the proposed proxy tasks, three learning methodologies have been tested including: instance contrastive learning (CLR), redundancy reduction (RR) and prototypical contrasting (PC).

**A. INSTANCE CONTRASTIVE LEARNING**

1) **PRELIMINARIES**

Given a dataset with \( N \) unlabeled elements \( D = \{x_1, x_2, \ldots, x_N\} \). The goal of representation learning is to learn the parameters \( \theta \) of a mapping function \( f_\theta(\cdot) \). It therefore maps the input samples into features vectors \( \{z_1, z_2, \ldots, z_N\} \), where \( z_i = f_\theta(x_i) \). The vector \( z_i \in \mathbb{R}^d \) captures the semantic content of \( x_i \), and its \( d \) components describes the data characteristics. In general, the mapping function \( f_\theta(\cdot) \) is composed of an encoder function followed by a projection network, usually thrown away in the downstream task.

Assume there is a stochastic data augmentation module \( T \) that transforms each sample \( x_i \) in the dataset \( D \) into two semantically-related, but different views samples, producing \( x_i', x_i'' \) where \((t, t') \sim T \). Hence, these two augmentations are treated as a positive pair considering that they relate to the same data example. Then, \((x_i', x_i'')\) are processed through the encoder function \( f_\theta(\cdot) \) to produce the features \((z_i', z_i'')\).

The contrastive loss, also called InfoNCE \cite{Chen2020}, is computed for a positive pair composed of the two augmentations \((x_i', x_i'')\) and a set of negative samples as follows:

\[
\ell(x_i', x_i'') = -\log \frac{\exp(s(z_i', z_i'')/\tau)}{\exp(s(z_i', z_i'')/\tau) + \sum_{k \in \mathcal{N}_i} \exp(s(z_i', z_i)/\tau)}
\]

We define \( s(u, v) = u^T v / ||u|| ||v|| \) as the dot product between \( l2 \)-normalized \( u \) and \( v \) (i.e cosine similarity) and \( \tau \) is the temperature parameter. The objective of this loss function is to discriminate its own augmented version \( x_i'' \) from a large number of other data samples \( \mathcal{N}_i \), where \( \mathcal{N}_i \) is the set the negative samples for \( x_i' \). It contains all the other samples in the batch as well as their augmented versions, that are different from its augmentation version \( x_i'' \). For a mini-batch of size \( B \), it is defined as:

\[
\mathcal{N}_i = \langle z_j', z_k' \rangle | \forall j \in [1, B] \text{ and } j \neq i, (t, t') \sim T \rangle
\]

The total loss is computed across all positive pairs, both \((x_i', x_i'')\) and \((x_i', x_i)\), in a mini-batch.

2) **CONTRASTIVE LOSS FOR AUDIO-VISUAL CORRESPONDENCE**

This section extends the contrastive loss for the audio-visual correspondence task. Consider a short video clip \( x_i \), we extract a sequence of audio frames \( x_i^a \) and a sequence of RGB frames \( x_i^v \). The RGB data is processed through \( g^s \circ f^v(\cdot) \) to generate the embedding vector \( z_i^v \). Similarly, the audio data \( z_i^a \) is fed to the audio network \( g^s \circ f^a(\cdot) \) to obtain the vector \( z_i^a \). This is well explained in Figure 1. These vectors are treated as two views of the same data, then the contrastive loss can be applied as defined before. For an audio sample \( z_i^v \), the corresponding RGB vector \( z_i^v \) is the positive example and vice-versa. The objective of the contrastive loss function is then
FIGURE 3. Cross-modal multi-task framework for audio-visual representation learning. It consists of three modules: Future prediction for audio modality, Future prediction for visual modality and Alignment between the two modalities.

FIGURE 4. Contrastive loss computation for audio-visual correspondence.

to distinguish the corresponding other modality vector from a large set of vectors containing both modalities’ vectors. The InfoNCE loss for the audio-visual correspondence task is defined as follows:

\[ \ell(x_i^a, x_i^v) = -\log \frac{\exp(s(x_i^a, x_i^v)/\tau)}{\exp(s(x_i^a, z_i^v)/\tau) + \sum_{k \in N_i} \exp(s(z_i^a, z_k)/\tau)} \]

(8)

In a mini-batch of size \( B \), the set of negative samples \( N_i \) for the audio sample \( x_i^a \) contains all the visual vectors \( z_j^v \) different from its corresponding visual vector \( z_i^v \), as well as all the audio vectors different from itself. Formally, it is defined as follows:

\[ N_i = \{ z_j^v | \forall j \in [1, B] \text{ and } j \neq i \} \cup \{ z_j^a | \forall j \in [1, B] \text{ and } j \neq i \} \]

The total loss is computed across all positive pairs, both \( (x_i^a, x_i^v) \) and \( (z_i^a, z_i^v) \), in a mini-batch. In [20], a memory bank is used to store the features \( \{ (x_i^a, x_j^v) \} \) for all the samples in the dataset which is very consuming for large-scale datasets. Instead, only the representations vectors of the current mini-batch \( (x_i^a, x_j^v) \) are needed to compute the loss function, and indeed reducing the memory space needed.

3) CONTRASTIVE LOSS FOR FUTURE PREDICTION

In the line of previous works on future prediction [42] and contrastive learning [3], this section presents the application of the contrastive framework for the future prediction task. As described in section III-A, the future prediction is an unimodal task that attempts to predict the next element of a sequential input in the latent space. In Figure 2, \( (z_i^m) \) is the representation of the predicted element of the sequence, while \( (z_i^{true}) \) is the representation of the true last frame of the sequence. Therefore, the vector \( (z_i^{true}) \) is considered as the positive sample for \( (z_i^{pred}) \) and vice-versa. The InfoNCE loss is computed to maximize the agreement between these two vectors, while pushing them further from the set of negative samples \( N_i \). It is defined as follows:

\[ \ell((z_i^m), (z_i^{true})) = -\log \frac{\exp(s((z_i^m), (z_i^{true}))/\tau)}{\exp(s((z_i^m), (z_i^{true}))/\tau) + \sum_{k \in N_i} \exp(s((z_i^m), z_k)/\tau)} \]

(9)

The total loss is computed across all positive pairs, both \( ((z_i^m), (z_j^m)) \) and \( ((z_i^m), (z_j^m)) \), in a mini-batch. The set of negative samples consists of both of the representations \( ((z_i^m), (z_j^m)) \) that are different from the sample itself. It also contains the representations of frames before aggregation \( h_t^m \) (\( t \) is the time index) in order to create more harder negatives.
The set $\mathcal{N}_i$ is defined as follows:

$$\mathcal{N}_i = \{(c^{p_j}_i)\forall j \in [1, B] \text{ and } j \neq i\} \cup \{(c^{p_j}_i)\forall j \in [1, B] \text{ and } j \neq i\} \cup \{(h^{t}_i)\forall j \in [1, B] \text{ and } t \leq T - 2\}$$

B. REDUNDANCY REDUCTION LOSS

Recently, [13] has proposed a self-supervised image representation learning, referred as Barlow Twins, that we extend to cross-modal data in this section. Following the same notation as in IV-A1, the principal idea is to compute the cross-correlation matrix $K$ between the batch outputs $z^i$ and $z^{i'}$. A single element of the matrix $K_{rc}$ is computed as follows:

$$K_{rc} = \frac{\sum_{t} z^i_{t,r} z^{i'}_{t,c}}{\sqrt{\sum_{t} (z^i_{t,r})^2 \sqrt{\sum_{t} (z^{i'}_{t,c})^2}}} \quad (10)$$

where $i$ is the index over the mini-batch dimension, and $(r, c)$ index the rows and columns of the square matrix $K$. Both vectors $z^i$, $z^{i'}$ are assumed to be normalized along the batch dimension, such as each component has a mean of zero and standard deviation equal to one.

Then, the Barlow Twins loss, also called redundancy reduction loss, is computed to encourage the cross-correlation matrix $K$ to be equal to the identity matrix. $\mathcal{L}$ is composed of two terms: invariance term and redundancy reduction term. The first one pushes the on-diagonal components $K_{rr}$ to be equal to 1, and thus making the embedding vectors invariant to the augmentations. The second term acts on the off-diagonal components to equate them to 0, and hence aims to decorrelate the components of the embedding vectors. It is defined as follows:

$$\mathcal{L} = \sum_{r} (1 - K_{rr})^2 + \lambda \sum_{r} \sum_{c \neq r} K_{rc}^2 \quad (11)$$

where $\lambda$ is a constant positive hyperparameter. It is a trade-off between the invariance term and the redundancy reduction term during training.

In order to solve the audio-visual correspondence task using the redundancy reduction, the two streams’ vectors $(z^i, z^{i})$ are treated as two views ‘augmentations’ of the same data. We, therefore, compute the cross-stream cross-correlation matrix between the outputs of the two encoders (Figure 1) along the batch dimension. It is defined as follows:

$$K_{rc} = \frac{\sum_{t} z^i_{t,r} z^{i'}_{t,c}}{\sqrt{\sum_{t} (z^i_{t,r})^2 \sqrt{\sum_{t} (z^{i'}_{t,c})^2}}} \quad (12)$$

The computation of the cross-stream cross-correlation matrix $K_{rc}$ matrix is not different from that of equation 10. It is computed between a batch of visual representations and that of audio representations. As previously mentioned, the vectors $(z^i, z^{i})$ are also assumed to be mean-centred along the batch dimension. Finally, the loss is computed using equation 11.

C. PROTOTYPICAL CONTRASTING

In instance contrastive learning explained in section IV-A1, the loss function is using explicit pairwise comparisons. This consequently consider all other samples in the batch as negatives that might share the same semantics of the positive sample. To mitigate this, we can use an online clustering algorithm and contrast different views to their cluster assignments instead of comparing features directly.

Using the same notation presented in section IV-A1, $(x^i, x^{i'})$ are two augmentations of $x_i$, and $(z^i, z^{i'})$ are produced using the mapping function. Let us define a set of $K$ trainable prototype vectors $C = \{c_1, \ldots, c_K\}$. Then, the features vectors $(z^i, z^{i'})$ are matched to $C$ to obtain their assignments $(q^i, q^{i'})$ respectively. The features and the assignments are used to define the following loss function:

$$\mathcal{L}(x^i, x^{i'}) = \ell(z^i, q^i) + \ell(z^{i'}, q^{i'}) \quad (13)$$

where $\ell$ describes the ability to predict the assignment of one view using the other view. Each term of the equation 13 represents the cross-entropy between the assignment and the probability obtained by applying the softmax on the similarity between $z$ and $C$:

$$\ell(z^i, q^i) = -\sum_{k} (q^i_k) \log \frac{\exp(s(z^i_k, c_k)/\tau)}{\sum_{k'} \exp(s(z^i_k, c_{k'}/\tau)} \quad (14)$$

where $\tau$ is the temperature parameter and $s$ is the similarity function defined previously.

The assignment procedure is done using an online clustering algorithm using only the feature vectors within a mini-batch. Following [12], [51], the samples in a mini-batch are constrained to be uniformly split across the prototypes in order to avoid the trivial solution. Given a mini-batch of $B$ feature vectors $Z = \{z_1, \ldots, z_B\}$ that need to be assigned to the prototypes $C = \{c_1, \ldots, c_K\}$ to obtain the assignments $Q = \{q_1, \ldots, q_B\}$, we optimize $Q$ using an Optimal Transport solver [52]:

$$\max_{Q \in \mathcal{Q}} \text{Tr} \left( Q^T C^T Z \right) + \varepsilon H(Q) \quad (15)$$

where $H(Q)$ is the entropy function of $Q$, and $\varepsilon$ is a parameter that adjusts the smoothness of the assignments. Following [12], the transportation polytope is restricted to the prototypes $C$.
to mini-batches:

\[ Q = \left\{ Q \in \mathbb{R}^{K \times B} \mid Q1_B = \frac{1}{K}1_K, Q^T1_K = \frac{1}{B}1_B \right\}. \]  
\( (16) \)

where \( 1_K \) indicates the vector of ones of dimension \( K \). Soft assignments \( Q^* \) are preserved as a solution of this problem, solved using the Sinkhorn-Knopp algorithm [53] and given as follows:

\[ Q^* = \text{Diag}(x) \exp \left( \frac{C^T Z}{\tau} \right) \text{Diag}(y), \]  
\( (17) \)

where \( x \) and \( y \) are two normalization vectors to ensure \( Q^* \) being a probability matrix. They are computed using a small number of iterations using the Sinkhorn-Knopp algorithm [53]. This algorithm is implemented on GPU and the number of iterations is fixed to 3, thus it is fast. Moreover, a queue mechanism is implemented to store the features from previous steps during the training as the batch size \( B \) is generally smaller than the number of prototypes \( K \).

We propose to apply the same idea for the audio-visual correspondence task. We extend the augmentation module by considering the RGB frames and the audio signal as two views of the data. The feature vectors are computed as explained in Section III-A to obtain the the two streams representations \( z_i^v, z_i^a \). For each modality, we define a set of \( K \) trainable prototypes: \( C_v = \{ c_1^v, \ldots, c_K^v \} \) for visual modality, and \( C_a = \{ c_1^a, \ldots, c_K^a \} \) for audio modality. Next, the features \( z_i^v, z_i^a \) are matched to the prototypes \( C_v, C_a \) respectively, to obtain the assignments \( q_i^v, q_i^a \) respectively as explained before. The loss for the tuple \( (x_i^v, x_i^a) \) is then defined as follows:

\[ L(x_i^v, x_i^a) = \ell(z_i^v, q_i^v) + \ell(z_i^a, q_i^a) \]  
\( (18) \)

where \( \ell(z_i^v, q_i^v) \) is given by:

\[ \ell(z_i^v, q_i^v) = -\sum_k (q_i^v)_k \log \frac{\exp(s(z_i^v, c_k^v) / \tau)}{\sum_{k'} \exp(s(z_i^v, c_{k'}^v) / \tau)} \]  
\( (19) \)

It indicates the ability of the model to predict the assignment of a representation from one modality using the embedding in the other modality. The term \( \ell(z_i^v, q_i^v) \) is defined the same way as in equation 19.

V. EXPERIMENTS AND RESULTS

A. DATASET

In the context of highlight detection, existing datasets include a wide range of videos depicting natural scenes such as TVSum [54] and SumMe [55]. These differ substantially to the images depicted in gaming videos. In the realm of datasets used for self-supervised learning from video, relatively few video datasets contain both visual and audio data.

For those reasons, a new dataset named VideoGame DB is created. Data was gathered from Twitch.tv\(^1\) and YouTube Gaming.\(^2\) Videos belong to gameplay of one specific game - Apex Legends - from random streamers were selected. Apex Legends game is a free-to-play hero shooter game where legendary competitors battle for glory, fame, and fortune on the fringes of the Frontier. It is available on PlayStation, Xbox, and PC. Since its release, Apex Legends has remained one of the highest-watched gaming categories on Twitch.tv.

Dataset creation consists of the several interrelated phases of content acquisition, data organisation and data processing. During the initial phase web scraping takes place to find suitable content. A list of Youtube and Twitch URLs are collected for the target game (Apex Legends). Then, the youtube-dl library\(^3\) is used to download these videos.

Instead of acquiring the entire video, non-overlapping sub clips of random two to five minutes were taken from these longer videos. This reduce the data size and ensure a certain diversity of the built dataset. In total, there are 6500 unique five minute video clips in the dataset, which are sampled from a total of 1132 unique longer videos. During self-supervised training, 10% of these video clips are randomly chosen as validation data used to monitor different metrics of the training process.

At the final phase, RGB frames were extracted from each of the videos at a rate of two frames per second. Human labelers manually labeled about 10000 of these frames. The labels indicated whether the frame was in-game (displaying gaming footage) or out-of-game (showed something else). Additional annotations were gathered for 6 binary features for the 1400 frames categorized as in-game, each indicating to the presence of some visual cue that occasionally appears during gaming footage. These include: damage marker, damage fx, hit marker, health bar depleting, knocking-down banner and killing banner. As described in Section V-B3, these labels were used as downstream tasks to assess the quality of the representations learned via the methods.

B. EXPERIMENTAL SETUP

1) DATA PREPROCESSING

For all experiments, RGB frames are extracted from videos at 2 FPS with spatial resolution of 224 × 224. For the

\(^1\)www.twitch.tv

\(^2\)www.youtube.com

\(^3\)https://github.com/ytdl-org/youtube-dl
audio stream, the audio clip is divided into non-overlapping 500 ms frames (the same frame-rate as RGB frames). Those frames are converted into time-frequency domain using the short-time Fourier Transform applying 25 ms windows every 10 ms. The resulting spectrogram is integrated into 64 mel-spaced frequency bins. The log operation on the magnitude of each bin is applied after adding a small offset for numerical stability. The resulting patches of size (50, 64) are the input for the audio encoder. The sequence length is fixed to 8 frames.

2) ARCHITECTURES
The ResNet-18 architecture [56] is used as an RGB frames encoder, while the VGGish architecture [57] is used for audio encoding as illustrated in Figure 8. The last fully connected layer of the VGGish is removed and the size of the last pooling layer was changed to be adequate with the input patch size. Recurrent neural networks (RNN) based on LSTM units are used for temporal aggregation. The LSTM hidden state has 512 dimensions.

Following SimCLR [3], a 2-layers MLP is used for all the projections heads when using contrastive learning to project the representations to 128 dimensional space. In the spirit of Barlow Twins [13], 3-layers MLP projectors are used, each with 8192 output units and a batch normalization operation after each linear layer, when using the redundancy reduction loss. In the case of prototypical contrasting, the projection MLP has just two hidden layers. A linear layer is added after this to go from the projection space to the clustering space. For downstream evaluation, only the ResNet-18 and VGGish backbone networks are used – all the others modules are removed.

3) DOWNSTREAM TASKS
The benefits of self-supervised pre-training is assessed based on the learned ResNet-18’s ability to classify the presence of visual cues in the human-annotated subset of extracted frames. The self-supervised pre-trained model, referred to as SSL ResNet-18, is compared to the same encoder with weights pre-trained on the ImageNet dataset [58], referred to as the baseline model. We also train the ResNet-18 on each task from scratch using random initialization in a supervised manner and we refer to this model as Random Init.

To evaluate the quality of the learned audio feature extractor, a subset of the data is labeled for voice activity detection (i.e., the detection of the presence or absence of human speech) using the VadNet model [59]. As this model produces a probability every 1-sec of audio, a linear interpolation of

![FIGURE 7. Visualization of some samples of the human labeled frames for different visual cues: damage marker, damage fx, hit marker, health bar depleting, knocking-down banner, killing banner.](image1)

![FIGURE 8. Detailed representation of the proposed architecture: ResNet-18 is the visual frame encoder, VGGish is the audio frame encoder. RNN layer is used to aggregate spatial features. MLP projection heads are used for each proxy task.](image2)
the obtained probabilities is applied to have a resolution of 0.5-sec that corresponds to the input length of the CNN encoder (VGGish). Then, the probabilities are thresholded on 0.5 to get the hard labels. The baseline model in that case is the same architecture VGGish pre-trained on the AudioSet dataset [60]. We also train the VGGish from random initialization in a supervised manner and we refer to this model as Random Init.

Moreover, two evaluation schemes for transfer learning are implemented: Linear evaluation named also pure transfer where the CNN encoder is only used for feature extraction and its weights are not updated during the backpropagation step, and Fine-tuning where a binary logistic regression with random initialization is added on the top of the CNN feature extractor and the whole network is fine-tuned.

4) METRIC EVALUATION
The area under the receiver operating characteristic (ROC) curve (AUROC) was used as the evaluation metric for all downstream tasks based on K-fold cross-validation technique with K = 2.

5) IMPLEMENTATION DETAILS
During the self-supervised pretraining, all models are trained for 5 epochs. Adam with LARS [61] was used as the optimizer. We use an initial learning rate of 0.002, with a cosine decay schedule [35], and a weight decay of 10^{-6}. Training is distributed across 4 NVIDIA V100 (32GB) GPUs using a batch size of 128. 0.1 is used for the temperature parameter \( \tau \), 300 for the number of prototypes \( K \), 5 \times 10^{-3} for the trade-off parameter \( \lambda \).

C. RESULTS
1) COMPARISON OF DIFFERENT LEARNING METHODOLOGIES FOR THE AVC TASK
This experiment compares the three learning methodologies used to solve the audio-visual correspondence: Instance contrastive learning, prototypical contrasting and redundancy reduction. Each model was trained as described in section V-B5 and the resulting representations are evaluated. Table 2 depicts the results of linear evaluation for each method and for each visual cue (task) using the AUROC metric. The column ImageNet refers to the CNN initialized with ImageNet weights (baseline model). We find that the three self-supervised methods outperform by far, on average, the baseline model (VGGish pretrained on AudioSet) is already powerful achieving 0.8875 in the linear evaluation. Even though, all the proposed approaches provide a non-negligible gain in terms of AUROC over the baseline method specially in the linear evaluation.

The quality of the resulting representations is also assessed in the audio embedding space to train efficiently the VGGish model. We find that the baseline model (VGGish pretrained on AudioSet) is already powerful achieving 0.8875 in the linear evaluation. Even though, all the proposed approaches provide a non-negligible gain in terms of AUROC over the baseline method specially in the linear evaluation.

2) COMPARISON OF DIFFERENT PROXY TASKS
In this section, we compare different configurations of the proposed pretext tasks. Using the contrastive loss, we compare training on the audio-visual correspondence task only, future prediction task only as well as combination of both tasks.

The weights of the visual frame encoder (ResNet-18) are obtained for these three configurations. Then, they are evaluated as previously explained on the visual cue detection downstream tasks. The results for linear evaluation are shown
TABLE 5. Results on visual downstream tasks for linear evaluation using different proxy tasks: AVC only, FP only and both. AVC: Audio-Visual only, FP: Future prediction only, AVC-FP: both tasks, ImageNet: Baseline.

| Visual Cue              | ImageNet | AVC   | FP    | AVC-FP |
|-------------------------|----------|-------|-------|--------|
| damage-marker           | 0.7485   | 0.8428| 0.7299| 0.9092 |
| damage-fx               | 0.8120   | 0.7785| 0.8205| 0.8440 |
| hit-marker              | 0.6860   | 0.3803| 0.6650| 0.8708 |
| health-bar-depleting    | 0.5679   | 0.7626| 0.7168| 0.8784 |
| knocking-down-banana     | 0.6823   | 0.7976| 0.5877| 0.8425 |
| killing-banner           | 0.6711   | 0.8103| 0.6919| 0.9152 |
| Average                 | 0.6950   | 0.8153| 0.6996| 0.8767 |

TABLE 6. Results on visual downstream tasks when fine-tuning Res-Net models based on different proxy tasks pretraining: AVC only, FP only and both. AVC: Audio-Visual only, FP: Future prediction only, AVC-FP: both tasks, ImageNet: Baseline.

| Visual Cue              | ImageNet | AVC   | FP    | AVC-FP |
|-------------------------|----------|-------|-------|--------|
| damage-marker           | 0.9223   | 0.9469| 0.9431| 0.9658 |
| damage-fx               | 0.8966   | 0.8745| 0.9138| 0.9971 |
| hit-marker              | 0.8072   | 0.8973| 0.8013| 0.8963 |
| health-bar-depleting    | 0.8718   | 0.9221| 0.9055| 0.9462 |
| knocking-down-banana     | 0.6936   | 0.9040| 0.6965| 0.8943 |
| killing-banner           | 0.8130   | 0.8966| 0.8856| 0.9451 |
| Average                 | 0.8337   | 0.9069| 0.8576| 0.9258 |

TABLE 7. Voice activity detection evaluation (AUROC) based on different proxy tasks pretraining: AVC only, FP only and both. AVC: Audio-Visual only, FP: Future prediction only, AVC-FP: both tasks, AudioSet: Baseline.

| Evaluation method | AudioSet | AVC   | FP    | AVC-FP |
|-------------------|----------|-------|-------|--------|
| Linear evaluation | 0.8875   | 0.9192| 0.8833| 0.9196 |
| Fine-tuning       | 0.9135   | 0.9204| 0.9094| 0.9228 |

in Table 5. On average, a significant performance improvement of 0.1817 AUROC points is obtained over the baseline model. Surprisingly, we find that using future prediction alone provides little to no improvement. However, when paired with the audio-visual task, the addition of the future prediction task yields significant gains.

Table 6 shows the results of fine-tuning evaluation of the three configurations. As before, the multi-task model is the best performing one, followed by the audio-visual model. The future prediction model is slightly better than the baseline model.

The obtained audio representations are also evaluated using the three methods, for voice activity detection as shown in Table 7. The same tendency that has been observed in the visual domain can also be seen here, but to a lesser extent.

3) UTILITY VS NUMBER OF LABELED SAMPLES
In this experiment, we study how the performance varies as the number of downstream training examples increases. For each trial, we sample a fixed and equal number of positive and negative samples for that task (the detection of one visual cue) to train on, while testing on the remaining examples.

To account for the randomness of sampling, each experience is repeated 10 trials and the average over the findings is taken as the performance. The number of labels we train on is dependent of the visual cue (task), as the number of positive samples is different in each task. For all experiments, at least 20% of the data is used for testing. The weights of the ResNet resulting of our best SSL method (combination of AVC and FP using contrastive learning) is compared to the baseline model (ImageNet initialization) as well as to random initialization.

Figure 10 shows the results in the case of linear evaluation. With a very limited number of training examples, self-supervised method has a significant utility for almost all the visual cues (tasks), although it remains constant for some tasks when labeled data grows. In Figure 9, the average of the AUROC metric across all the visual tasks is shown for different training set sizes. With just 20 labels per class, the model achieves very well over 0.8 average AUROC, while the ImageNet pre-training is closer to 0.6.

Figure 11 shows the results of the same experiment in the case of fine-tuning. We still have an important utility of the SSL pre-training for most tasks. However, the gap to the baseline model narrows. Figure 9b shows the average of those results across all the visual tasks indicating the same tendency seen before.

4) HOW MANY UNLABELED VIDEOS DO WE NEED TO LEARN GOOD FEATURES?
It is also worthwhile to investigate the quality of the learnt representations in relation to the amount of unlabeled videos in the self-supervised step. In Table 8, we report the average performance across all visual downstream tasks for different dataset sizes. Those results correspond to training the AVC task using contrastive learning by varying the size of the dataset. The quality of the representations tends to improve with the size of the dataset in the linear evaluation. In addition, the gain of performance over the baseline model seems significant even with a very small dataset of 100 video clips. When fine-tuning the obtained models, we observe that the performances for different sizes are very comparable to each other. Similar to linear evaluation, training on a small size of 100 video clips gives less performance but still outperforms the baseline model.

5) WHICH LAYER OF THE CNN IS MOST USEFUL FOR DOWNSTREAM TASK?
This experiment investigates which layer is most useful for the downstream tasks. As the ResNet model is only a part of the whole learning system, it is not clear which layer to take as a representation of the RGB frames. We tested two
intermediate layers of the ResNet-18, \textit{conv3}_2 the output of 3\textsuperscript{rd} conv block of dimension 128, and \textit{conv4}_2 the output of 4\textsuperscript{th} conv block of dimension 256 (we use the same notation used in the original paper for different layers, for more details see [56]). Output Layer is the last layer before the classification layer of the ResNet-18, which contains 512 neurons. During SSL pre-training, we used a feature projection module (linear layer followed by layer a normalization and then an activation function) between the output of the ResNet-18 and the input of the RNN temporal encoder. Thus, we tested also the representations resulting of the feature projection module noted as Projected Features. It has also 512 units. Table 9 presents the results of this experiment in the case of linear evaluation. The output layer is by far the best representations of frames and outperforms the projected features.

We can think of the feature projection module as a domain adaptation when aggregating the spatial vectors. Table 10 reports the results of this experiment in the case of fine-tuning evaluation. On average, the output layer produce the best representations of the RGB frames.

6) APPLICATION TO A DIFFERENT VIDEOGAME

In this last experiment, we extend our analysis to an additional videogame: PlayerUnknown’s Battlegrounds (PUBG) Mobile. PUBG is a player versus player shooter game in which up to one hundred players fight in a battle royale. We apply the same method explained in section V-A to build another dataset for this game resulting in a set of 5914 video clips of two to five minutes. Similarly to the presented Video Gameplay DB, human labelers manually annotated about 32000 frames extracted from those video clips, where they were categorized as in-game or out-game. Additional labels were gathered for 6 binary characteristics of the 8800 frames labeled as in-game, each indicating the presence of some visual cue that occasionally appears during gaming footage. These include: damage marker, damage fx, knocked down health bar, killing message, red kill feed entry and blue kill feed entry.

The same pipeline is applied to this dataset in terms of the data preparation and data preprocessing steps. We train the proposed framework (explained in section III-C) using contrastive learning (section IV-A). The benefits of self-supervised pre-training is assessed based on the performance of the learned ResNet-18 to detect the presence of visual cues in the human-annotated subset.

Table 11 shows the results of linear evaluation of the resulting CNN for each visual cue (task) using the AUROC metric. The column \textit{ImageNet} refers to the performance of the CNN initialized with ImageNet weights (baseline model). We find
that the proposed self-supervised approach outperforms by far, on average, the baseline model.

Finally, Figure 12 shows the superiority of the proposed pre-training method (in the linear evaluation context) when only a very limited number of training examples are available. With just 20 labels per class, the model achieves well over average 0.75, while ImageNet pre-training is closer to 0.6.

D. DISCUSSIONS
To further support our hypothesis, SimCLR [3] is trained on the images of Videogame DB dataset in isolation. Even after iterating over different augmentation setups (which must be adjusted to the domain), the performance on the linear evaluation was worse than that of the pretrained ImageNet model. This is likely due in large part to the fact that it is difficult to find image-only augmentations that would occur
in real data (for instance, random cropping is not something that will ever occur in videogame footage), but it also shows that image-level self-supervised algorithms are unlikely to work in this context. Moreover, the video component is effective, as opposed to an ‘only’ approach, as the visual cues that need to be detected in the downstream evaluation are events that are often lagging or leading indicators of other visual cues (or linked to contemporaneous audio events).

As the baseline visual encoder is the ResNet-18 model pre-trained on ImageNet, the question of whether ImageNet features can transfer well to other domains may be raised. For instance in [62], [63], one can see that the features detected by CNNs trained on Imagenet can be quite abstract, and include
textures and shapes. As a result, we can quite easily imagine these features still being useful in the video games context.

VI. CONCLUSION AND FUTURE WORK

This paper introduces a high-performing self-supervised learning approach for video games highlight detection based on a cross-modal multi-task framework and contrastive learning. The proposed methods show that the self-supervised pre-training on unlabeled videos can produce models that outperform by far baselines on downstream tasks. It gives a significant reduction in the amount of labeled data needed to get the same performance using the baselines. We have also demonstrated that the amount of video clips used to learn the feature space can be small and still outperforms the baseline model. Finally, we demonstrated that the proposed framework can also help for another distribution of the dataset (a new game), which confirms that it can be generalized to other datasets. In future work, we plan to evaluate the performance of the proposed methods on different video datasets from other domains, and to devise and evaluate alternative proxy tasks and loss function formulations.

ACKNOWLEDGMENT

Powder is making the camera of the metaverse and applies artificial intelligence to highlights detection in videogames, and emotion understanding (https://powder.gg/).

REFERENCES

[1] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidic, and L. Farhan, “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions,” J. Big Data, vol. 8, no. 1, p. 53, Mar. 2021.

[2] L. Jing and Y. Tian, “Self-supervised visual feature learning with deep neural networks: A survey,” 2019, arXiv:1902.06162.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” 2020, arXiv:2002.05709.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013, arXiv:1301.3781.

[5] J. Pennington, R. Socher, and C. Manning, “GloVe: Global vectors for word representation,” in Proc. Conf. Empirical Methods Natural Lang. Process., Doha, Qatar, Oct. 2014, pp. 1532-1543. [Online]. Available: https://aclanthology.org/D14-1162.

[6] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” 2016, arXiv:1607.04606.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.

[8] T. B. Brown et al., “Language models are few-shot learners,” 2020, arXiv:2005.14165.

[9] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” 2016, arXiv:1603.09246.

[10] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” 2018, arXiv:1807.05520.

[11] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” 2021, arXiv:2104.14294.

[12] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” 2020, arXiv:2006.09882.

[13] J. Zhontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, “Barlow twins: Self-supervised learning via redundancy reduction,” 2021, arXiv:2103.03230.

[14] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui, “Spatiotemporal contrastive video representation learning,” 2020, arXiv:2008.03800.

[15] Y. Yao, C. Liu, D. Luo, Y. Zhou, and Q. Ye, “Video playback rate perception for self-supervised spatio-temporal representation learning,” 2020, arXiv:2006.11476.

[16] B. Fernando, H. Bilen, E. Gavves, and S. Gould, “Self-supervised video representation learning with odd-one-out networks,” 2016, arXiv:1611.06646.

[17] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran, “Self-supervised learning by cross-modal audio-video clustering,” 2019, arXiv:1911.12667.

[18] M. Toering, I. Gatoopoulou, M. Stol, and V. Tao Hu, “Self-supervised video representation learning with cross-stream prototypical contrasting,” 2021, arXiv:2101.10137.

[19] T. Han, W. Xie, and A. Zisserman, “Self-supervised co-training for video representation learning,” 2020, arXiv:2010.09709.

[20] P. Morgado, N. Vasconcelos, and I. Misra, “Audio-visual instance discrimination with cross-modal agreement,” 2020, arXiv:2004.12943.

[21] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, no. 5786, pp. 504-507, 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/sites/entrez?db=pubmed&ai=16873662&cmd=showdetailview&index=google

[22] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, “Building high-level features using large scale unsupervised learning,” 2011, arXiv:1112.6209.

[23] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proc. NIPS, 2014, pp. 2672–2680.

[24] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” 2016, arXiv:1604.07379.

[25] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” 2016, arXiv:1603.08511.

[26] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” 2015, arXiv:1505.05192.

[27] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” 2018, arXiv:1803.07729.

[28] J. Xu, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” 2015, arXiv:1511.06355.

[29] J. Yang, D. Parikh, and D. Batra, “Joint unsupervised learning of deep representations and image clusters,” 2016, arXiv:1604.03628.

[30] A. Dosovitskiy, P. Fischer, J. Tobias Springenberg, M. Riedmiller, and T. Brox, “Discriminative unsupervised feature learning with exemplar convolutional neural networks,” 2014, arXiv:1406.6909.

[31] Z. Wu, X. Xiong, S. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance-level discrimination,” 2018, arXiv:1805.01978.

[32] M. Gutmann and A. Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in Proc. 13th Int. Conf. Artif. Intell. Statist., Y. W. Teh M. Titterington, Eds., vol. 9. Sardinia, Italy: Chia Laguna Resort, May 2010, pp. 297–304. [Online]. Available: https://proceedings.mlr.press/v9/gutmann10a.html

[33] X. Chen and K. He, “Exploring simple Siamese representation learning,” 2020, arXiv:2011.16566.
[34] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, “TVSum: Summarizing video content,” 2016, arXiv:1609.02612.

[35] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” 2016, arXiv:1609.02612.

[36] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, “Tracking emerges by colorizing videos,” 2018, arXiv:1806.09594.

[37] A. Jansen, Y. Song, J. Vallmitjana, and A. Stent, “Self-supervised learning: Unsupervised learning using temporal order verification,” 2016, arXiv:1601.08561.

[38] D. Kim, D. Cho, and I. So Kweon, “Interactive video representation learning with space-time cubic puzzles,” 2018, arXiv:1811.09795.

[39] S. Jenn, G. Meishvili, and P. Favaro, “Video representation learning by recognizing temporal transformations,” 2020, arXiv:2007.10730.

[40] N. Srivastava, E. Mansimov, and R. Salakhutdinov, “Unsupervised learning of video representations using LSTMs,” 2015, arXiv:1502.04681.

[41] F. Vondrick, F. Agrawal, V. Koltun, and L. Van Gool, “FlowNet: Optical flow from convolutional networks,” 2015, arXiv:1504.06852.

[42] N. Sayed, B. Brattoli, and B. Ommer, “Cross and learn: Cross-modal self-supervised learning,” 2018, arXiv:1811.03879.

[43] B. Korbar, D. Tran, and L. Torresani, “Cooperative learning of audio and video models from self-supervised synchronization,” 2018, arXiv:1807.00230.

[44] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, “Perceiver: General perception with iterative attention,” 2021, arXiv:2103.03206.

[45] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, “Decomposing motion and content for natural video sequence prediction,” 2017, arXiv:1706.08033.

[46] M. Mathieu, C. Couprie, and Y. LeCun, “Deep multi-scale video prediction,” 2015, arXiv:1511.05440.

[47] A. Piergiovanni, A. Angelova, and M. S. Ryoo, “Evolving losses for unsupervised video representation learning,” 2020, arXiv:2012.1777.

[48] K. Sohn, “Improved deep metric learning with multi-class n-pair loss objective,” in Proc. Adv. Neural Inf. Process. Syst., vol. 29, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016. [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/b6180037abbebea991d8b1b232f8a8ca9-Paper.pdf

[49] Y. Markus Assano, C. Rupprecht, and A. Vedaldi, “Self-labelling via simultaneous clustering and representation learning,” 2019, arXiv:1911.05171.

[50] G. Peyré and M. Cuturi, “Computational optimal transport: With applications to data science,” Found. Trends Mach. Learn., vol. 11, nos. 5–6, pp. 355–607, 2019, doi: 10.1561/2200000073.

[51] M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transport,” in Proc. Adv. Neural Inf. Process. Syst. vol. 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2013. [Online]. Available: https://proceedings.neurips.cc/paper/2013/file/2a1d0e07db2c27e13572cbf59eb343d-Paper.pdf

[52] Y. Song, J. Vailmitjana, A. Stent, and A. Jaimes, “TVSum: Summarizing web videos using titles,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 5179–5187.

[53] M. Gogli, H. Grabner, H. Riemenschneider, and L. Van Gool, “Creating summaries from user videos,” in Proc. ECCV, 2014, pp. 505–520.

[54] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015, arXiv:1512.03385.

[55] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. Channing Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, “CNN architectures for large-scale audio classification,” 2016, arXiv:1609.09430.

[56] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255. [Online]. Available: https://journals.ieee.org/abstract/document/5206848/

[57] J. Wagner, D. Schiller, A. Seiderer, and E. André, “Deep learning in paralinguistic recognition tasks: Are hand-crafted features still relevant?” in Proc. Interspeech, Hyderabad, India, Sep. 2018, pp. 147–151.

[58] HACENE TERBOUCHE received the Engineering Diploma in electronics and computer science from the Ecole Nationale Polytechnique, Algiers, Algeria, in 2020, and the M.S. degree in data science and artificial intelligence from the Université de Paris-Saclay, Paris, France, in 2021. He is currently pursuing the Ph.D. degree in computer vision with Université Paris-Est, Créteil, France. His research interests include self-supervised learning, multimodal learning, and video highlighting.

[59] LIAM SCHONEVELD received the Bachelor of Commerce (Liberal Studies) degree from the University of Sydney, in 2013, and the M.Sc. degree in artificial intelligence from the University of Amsterdam, in 2017. He is currently a Principal AI Researcher at Powder. His research interests include computer vision and deep learning, with recent work published in the field of emotion recognition. His current research interests include self-supervised learning and neural network optimization.

[60] OSISIN BENSON received the B.A. degree in theoretical physics from the Trinity College Dublin, Ireland, in 2016, and the M.Sc. degree in data science from The University of Edinburgh, U.K., in 2017. He is currently at Powder, his research interests over his career have included machine learning, computer vision, generative models, and self-supervised learning.

[61] ALICE OTHMANI has been an Associate Professor with Université Paris-Est, Créteil, since 2017. Her research interests include developing computer vision and artificial intelligence solutions for healthcare. She has skills and several experiences in computer vision, machine learning, and signal and image processing projects. She has been working in several international institutions, such as the Ecole Normale Supérieure de Paris, Collège de France, and Agency for Science, Technology and Research (A*STAR), Singapore.