

# Unsupervised Video Continual Learning via Non-Parametric Deep Embedded Clustering

Nattapong Kurpukdee  
nattapong.kurpukdee@york.ac.uk

Adrian G. Bors  
adrian.bors@york.ac.uk

Department of Computer Science,  
University of York,  
York, YO10 5GH, UK

## Abstract

We propose a realistic scenario for the unsupervised video learning where neither task boundaries nor labels are provided when learning a succession of tasks. We also provide a non-parametric learning solution for the under-explored problem of unsupervised video continual learning. Videos represent a complex and rich spatio-temporal media information, widely used in many applications, but which have not been sufficiently explored in unsupervised continual learning. Prior studies have only focused on supervised continual learning, relying on the knowledge of labels and task boundaries, while having labeled data is costly and not practical. To address this gap, we study the unsupervised video continual learning (uVCL). uVCL raises more challenges due to the additional computational and memory requirements of processing videos when compared to images. We introduce a general benchmark experimental protocol for uVCL by considering the learning of unstructured video data categories during each task. We propose to use the Kernel Density Estimation (KDE) of deep embedded video features extracted by unsupervised video transformer networks as a non-parametric probabilistic representation of the data. We introduce a novelty detection criterion for the incoming new task data, dynamically enabling the expansion of memory clusters, aiming to capture new knowledge when learning a succession of tasks. We leverage the use of transfer learning from the previous tasks as an initial state for the knowledge transfer to the current learning task. We found that the proposed methodology substantially enhances the performance of the model when successively learning many tasks. We perform in-depth evaluations on three standard video action recognition datasets, including UCF101, HMDB51, and Something-to-Something V2, without using any labels or class boundaries.

## 1 Introduction

Unsupervised Continual Learning (UCL) aims to progressively learn from unlabeled data by finding associations based on certain criteria while addressing catastrophic forgetting. Usually, groupings of data are made according to their statistical similarity. A key challenge to this process is that of being able to preserve what was learned in the past, representing the stability, while also having the ability to learn novel information, corresponding to plasticity. The trade-off between stability and plasticity in unsupervised video learning represents a challenging endeavor.

Most unsupervised class-incremental learning approaches developed for the image domain [9, 12, 21, 22, 36, 45] focus on aligning unlabeled data with those from categories derived from labeled source data. However, these methods rely on large supervised models and make unrealistic assumptions such that all given unlabeled data represent novel information, without considering overlaps with previously learned data. Furthermore, they require predefined cluster boundaries, which limit their applicability to real-world scenarios.

Unlike image-based UCL models, the video domain received very limited attention in continual learning studies. In real-world situations, incoming unlabeled data often consists of data sourced from different probabilistic representations, corresponding to mixed sets of data categories overlapping with each other as well as with the previously learned information. In such cases, a challenge is represented by the insufficient amount of data available to fully train the model and by the differences in the amount of such data from different categories. In the method proposed here we initially extract feature sets using a video transformer [43]. Then, we successively organize the extracted sets of features into clusters, representing the statistical distribution characterizing the learned data. During the learning of each task, the model continuously associates new feature sets with existing clusters while also creating new clusters according to a novelty criterion, optimizing both memory and time efficiency. We consider a non-parametric clustering method by adapting the mean-shift algorithm [8, 11, 9] as a continual representation through the Kernel Density Estimation (KDE) representation of video data.

In this paper, we address the real-world challenge of unsupervised continual learning for video, where neither task boundaries nor class boundaries are provided to the learner. We propose a simple yet effective and practical approach entitled the unsupervised Video Continual Learning based on Kernel Density Estimation (uVCL-KDE).

Our main contributions are as follows :

- We explore a non-parametric continual learning setting through the proposed uVCL-KDE, by grouping data based on their kernel-density representation affinities.
- We propose to use the mean-shift method, for defining sets of clusters when applying uVCL-KDE on video features in a continual learning setting. We also extend to the uVCL-KDE-RBF by adding a linear mapping on top of the clustering, as in the Radial Basis Function (RBF) networks.
- We introduce a benchmark evaluation protocol to facilitate a realistic assessment framework for future research and provide an extensive experimental analysis of the effectiveness of each component in our proposed approach.

## 2 Related Work

Continual or lifelong learning is characteristic to all living beings allowing them to adapt in various life situations. However, AI systems suffer from catastrophic forgetting when they are retrained on new datasets and have a very low probability of fulfilling the tasks learned in the past. In this section, we begin by reviewing existing research on supervised continual learning, highlighting key challenges. Following this, we examine various unsupervised continual learning and their potential for real-world applications.

## 2.1 Supervised Continual Learning

In Supervised Continual Learning (SCL), models are trained sequentially on a series of  $k$  tasks  $\{\tau_0, \tau_1, \dots, \tau_k\}$ , where each task involves learning a set of data and its corresponding labels. SCL approaches are generally categorized into those based on regularization, architecture expansion and memory-based methods. Regularization-based approaches use some specific terms in the loss function in order to reduce catastrophic forgetting [10, 17]. Meanwhile, expansion architecture models add new neurons, layers or entire modules in order to enable the learning of new tasks [14]. Memory-based methods mitigate catastrophic forgetting by retaining a limited subset of training data from previously learned tasks  $\{\tau_0, \tau_1, \dots, \tau_{k-1}\}$  in a memory buffer. Then they draw samples from the memory buffer when learning a new task  $\tau_k$ .

Most existing video supervised continual learning (VSCL) models are adaptations of methods initially developed for image continual learning [10, 9, 11, 18, 28, 32, 33, 35, 44, 46, 47, 49, 50]. Some image-based VSCL approaches, such as the Incremental Classifier and Representation Learning (iCaRL) [35], and Bias Correction (BiC) [47], have been directly extended to the VSCL models, [25, 29, 42]. Most models use memory buffers to store videos from previously learned classes during continual learning, together with their labels, aiming to address catastrophic forgetting [20]. Other methods, specifically proposed for video SCL [25, 27, 29, 33, 42], focus on mitigating forgetting in video-based tasks by using memory buffers or prompts to retain the knowledge of previously learnt classes. Many video SCL models rely on Convolution Neural Networks (CNNs) as their backbones. Recently, a promising direction of research is represented by the integration of large language models (LLMs) and vision models for video SCL [53, 41]. However, these methods are still limited in their real-world applicability due to their reliance on costly human annotation and labeling.

## 2.2 Unsupervised Continual Learning

Unsupervised learning in the image domain is a rapidly growing research area, with various methods [9, 48] leveraging visual features for learning without any labeled information. Similarly, the unsupervised learning in videos is increasingly gaining attention, with Zhuang *et al.* [51] introducing a two-pathway approach for unsupervised video learning. Meanwhile, unsupervised Continual Learning (UCL) in the image domain has been explored in several studies, such as [6, 13, 22, 26, 34, 37, 39, 48]. In these models, pseudo-labels are used to replace human annotations for learning new, non-overlapping categories of image data. To address the problem of catastrophic forgetting, some methods use the Deep Generative Replay (DGR) replay training. Additionally, simple classifiers like K-Nearest Neighbors (KNN) are employed in the latent space for unsupervised data assignment.

The field of unsupervised continual learning in video domain remains under-explored. While unsupervised domain adaptation has been studied for both images and videos in various applications [9, 12, 21, 36, 45], these models typically depend on a pre-trained, supervised source model, while adapting the unsupervised target data to the identified primary source representations.

Unlike the previous studies that consider either class or category incremental learning settings, in this paper we propose a simple yet effective framework, consisting of learning from the data of multiple mixed categories in an unsupervised way. The proposed approach relies on the kernel density-based data representation in the video feature space. The result-  
ing peaks in the non-parametric representation provided by the Kernel Density Estimation

(KDE) of the feature space are used as data representation attributes to store and then replay past information during continual learning.

### 3 Problem Setup

In this paper, we study unsupervised video continual learning, aiming to learn and structure a data space  $\mathcal{H}$ , given a sequence of  $K$  tasks,  $\{\tau_1, \tau_2, \dots, \tau_K\}$ . During each task  $\tau_k$ ,  $k = 1, \dots, K$  a set of video data is provided, denoted as  $\tau_k = \{\mathbf{v}_i\}_{i=1}^{n_k}$ , where  $\mathbf{v}_i$  represents  $i$ -th video sample and  $n_k$  is the number of videos to be learnt during task  $\tau_k$ , with a total number of training data as  $n = \sum_{k=1}^K n_k$ . In this study we consider the challenging situation of learning unsupervised tasks, where there are no labels for the training set. We also assume no pre-defined structure or categorization in the video data. Our objective is to define a labeling function  $f(\cdot)$ , parameterized by a deep learning network, that assigns pseudo-labels to the data,  $\tilde{y}_i^j = f(\mathbf{v}_i)$ , where we assume that the ideal label, unknown to  $f(\cdot)$  is  $y_i^j$ , where  $j$  is the label's identifier. The pseudo-labels assignment is performed according to the video's feature properties and the characteristics of the labeling function  $f(\cdot)$ .

Under the most general, yet realistic, setting, each task  $\tau_k$  consists of mixed class/category data, corresponding to a mixture of distributions, where  $\tau_k = \{\mathbf{v}_i, y_j | i = 1, \dots, n_k, j = 1, \dots, m_k\}$ , where  $y_j$  represents true labels, unknown in the unsupervised learning system  $f(\cdot)$  and  $m_k$  is the number of data categories provided at  $\tau_k$ . The data learnt at task  $\tau_k$  may or may not overlap statistically with previously learned data, without actually providing explicit class boundaries for any learnt datum. This presents a more realistic, yet challenging problem, that was not tackled before, as the model must learn at each step from completely unstructured data while aiming to form semantically meaningful data associations defined by  $f(\cdot)$ . Our aim is to create a series of representative clusters, with each cluster characterized by a pseudo-label  $\tilde{y}_i^j$ ,  $j = 1, \dots, l_k$ , where  $l_k$  represents the number of identified clusters at task  $\tau_k$ .

In the proposed unsupervised continual learning, in order to mitigate catastrophic forgetting, we consider storing a small set of video features  $\mathcal{P}_k = \bigcup_{j=1}^{l_k} \{\mathbf{v}_i, \tilde{y}_j\}_{i=1}^{n_k}$ , as exemplars associated with each cluster  $j$ , defined at task  $\tau_k$ . These data are then reused during each subsequent task learning, while more clusters are added. Unlike supervised class-incremental learning approaches [0, 6, 10, 18, 28, 32, 33, 35, 42, 44, 46, 47, 49, 50], which operate with fixed class increments,  $l_1 = l_2 = \dots = l_K$ , our approach does not require to predefine the number of classes for each incremental step. Instead, the number of classes dynamically increases throughout the continual learning process, reflecting the non-stationary nature of the real-world data.

### 4 Method

We propose a simple yet effective method for unsupervised video continual learning, based on non-parametric deep-embedded cluster assignments. The overall procedure for unsupervised continual learning of a sequence of video tasks  $\{\tau_1, \tau_2, \dots, \tau_K\}$  is outlined in Fig. 1. In the following, we describe the proposed approach, which includes feature extraction using a video transformer, non-parametric deep cluster embedding along the continual learning process and the memorization of features for the memory replay during future task learning.

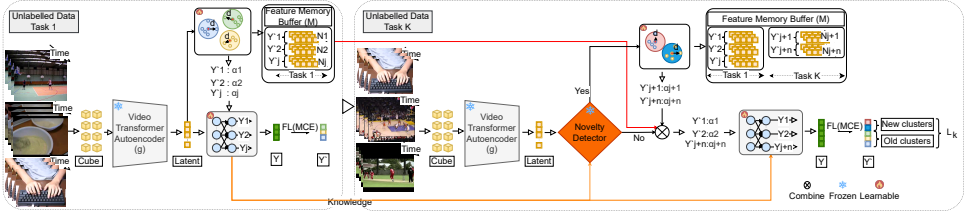


Figure 1: Overview of the proposed unsupervised video continual learning based on the Kernel Density Estimation (uVCL-KDE).

## 4.1 Feature extraction

Each task involves using a video auto-encoder transformer network for extracting video features,  $\{\mathbf{x}_{k,i} = g(\mathbf{v}_i)\}_{i=1}^{n_k}$ , using a pre-trained auto-encoder video transformer network, considered of size  $|\mathbf{x}_{k,i}|$ , for any task  $\{\tau_k | k = 1, \dots, K\}$  and  $i = 1, \dots, n_k$ . The network  $g(\cdot)$  is used to extract the features when learning all tasks  $\{\tau_k | k = 1, \dots, K\}$ , without being retrained, thus ensuring a consistent feature space over the entire data space. These features are then grouped into a number of clusters through a deep clustering algorithm, described in the following.

## 4.2 KDE-based Deep Embedded Clustering

In this paper we propose the unsupervised Video Continual Learning using Kernel Density Estimation (KDE), namely uVCL-KDE. The proposed methodology relies on the online non-parametric deep embedded clustering strategy. The video feature data are organized by the Mean-shift [3, 4, 5], which is a dynamic data-representation KDE-based clustering method which does not require to know the number of clusters. The data representation in the KDE is given by considering a kernel function centered on each sample and calculating the resulting probability density function (pdf) :

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n_k} \mathcal{K}(\mathbf{x} - \mathbf{x}_i) = \sum_{i=1}^{n_k} \mathcal{K}\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h^2}\right) \quad (1)$$

where  $\mathbf{x}_i$ ,  $i = 1, \dots, n$  represents the feature vector of the input data, where we consider  $\mathcal{K}$  depending on a bandwidth  $h$ , with each kernel centered at a data sample.  $h$  can influence the number of peaks in the resulting probability density function (pdf) representation [3].

When considering the Gaussian kernel, with its cluster center  $\mu_j$  as the kernel center while  $h$  corresponds to the standard deviation, the Mean-shift, adaptively moves its cluster centers towards the peaks in the pdf representation, like the one from Eq. (1). Then, when considering  $\mathcal{K}$  as a Gaussian kernel [3] in Eq. (1) and differentiating this pdf representation, we can iteratively calculate the mean-shift as :

$$M(\mu_j^t) = \frac{\sum_{i=1}^{n_k} \left( -\mathbf{x}_i \frac{\|\mu_j^t - \mathbf{x}_i\|^2}{2h^2} \right)}{\sum_{i=1}^{n_k} \left( -\frac{\|\mu_j^t - \mathbf{x}_i\|^2}{2h^2} \right)} - \mu_j^{t-1}, \quad (2)$$

where  $\mu_j^t$  is the cluster center found at the  $t$ -th iteration of the Mean-shift algorithm. The Mean-shift is iteratively used to update the mean as  $\mu_j^{t+1}$ , and then is recalculated until the cluster centers are found  $\mu_j = \mu_j^t$  when  $\mu_j^t \approx \mu_j^{t-1}$ . After finding the peaks from the KDE representation corresponding to all the data from the given task  $\{\tau_k | i = 1, \dots, n_k\}$  two cluster candidates are considered as distinct if there is a local minima on the line that joins them, while otherwise the two clusters are merged.

Eventually, a cluster is associated with each peak in the resulting KDE and the peaks are found through the mean-shift, as described above. Consequently, data are associated with the peaks and clusters. In order to avoid forgetting in the unsupervised video continual learning, after learning each task, a certain number of video features, are stored in memory buffers in order to be used for future training. We assign a memory buffer  $\mathcal{M}_j$ ,  $j = 1, \dots, l_k$  for each peak of the KDE, considered as defining a cluster in the KDE representation. When proceeding to the next task  $\tau_{k+1}$ , all the data from the memory are combined and used together with the new data provided with the task, forming an updated KDE landscape. After iterating through equations (1) and (2) new clusters are formed when novel data are identified, resulting in a probabilistic representation that adapts to the novel data, while also preserving the knowledge accumulated during the learning of all tasks  $\{\tau_1, \tau_2, \dots, \tau_k, \tau_{k+1}\}$ .

### 4.3 Linear cluster self-allocation

All video data  $\mathbf{x}_{k,i}$  associated uniquely to each cluster, are assigned with a pseudo-label  $\tilde{y}_{k,i}$ . Such data allocations, defined by the centers  $\mu_{k,j}$ , can be seamlessly integrated with regularization-based methods, such as knowledge distillation loss or other approaches. In this context, a multi-class cross-entropy loss is applied to learn the cluster assignments for the data  $\mathbf{x}_{k,i}$ , with such pseudo-labels being akin to labels typically used in supervised settings. Here,  $\tilde{y}_j$ ,  $j = 1, \dots, L_K$  represent the  $L_K$ -cluster assignments and these are used as targets within a linear classifier, like in a Radial Basis Functions (RBF) network [20]. The cluster assignments  $\tilde{y}_j$  are used as training labels in a multi-class classification task. The classifier then outputs class probabilities using softmax normalization, as in the following :

$$\sigma(\tilde{y}_{i,j}) = \frac{\exp(\tilde{y}_{i,j})}{\sum_{j=1}^{L_K} \exp(\tilde{y}_{i,j})} \quad \text{for } i = 1, 2, \dots, n, \quad (3)$$

where,  $\tilde{y}_{i,j}$  is the vector of raw outputs from the neural network, and  $\sigma(\tilde{y}_{i,j})$  is the softmax output corresponding to the probability that the input belongs to class  $i \in L_K$  and  $L_K$  is the number of pseudo-classes, given by the number of clusters.

This method, which trains a linear layer on top of the clusters inferred from the KDE representation, akin to the Radial Basis Function (RBF) Networks [20], is named uVCL-KDE-RBF. For training the last layer of uVCL-KDE-RBF, we use the multi-class cross-entropy (MCE) loss :

$$MCE = - \sum_{j=1}^{L_K} \bar{y}_{o,j} \log(p_{o,j}), \quad (4)$$

where  $L_K$  represents the classes defined by the pseudo-clusters,  $\bar{y}$  is a binary indicator of 0 or 1 indicating whether the class label  $j$  is the correct classification label for observation  $o$ .  $p$  is the predicted probability observation  $o$  for the class  $j$ .

A challenging aspect in the unsupervised continuous learning is the presence of imbalance in the amount of different data categories. Consequently, we address such imbalances

during training by employing the Focal Loss for weighting the contribution of each cluster [23]. We then modify the MCE using Focal Loss (FL) from [23], by replacing the classes with the pseudo-labeled clusters, as :

$$FL(MCE) = \alpha_j * (1 - \exp(-MCE))^\gamma * MCE, \quad (5)$$

where  $\alpha_j$  is the pseudo-cluster balance weight,  $j = 1, \dots, L_K$ , and  $\gamma = 2$  is a modulating factor for the multi-class cross-entropy loss.

#### 4.4 Novelty detector and cluster augmentation

New clusters are defined when the new data, provided in the tasks from the sequence being learned, indicates completely different information from that already known by the uVCL-KDE, according to :

$$\arg \min_{j=1}^{L_{K-1}} d_j(\mathbf{x}_{k,i}) = \|g(\mathbf{x}_{k,i}) - \mu_{k,j}\| > \Theta_1, \quad (6)$$

where  $\Theta_1$  is a threshold defining new clusters. We estimate  $\Theta_1$  using the data from the first task, as the maximum of the distances between each two existing cluster centers.

In the case of the uVCL-KDE-RBF we consider the probability  $\sigma(\tilde{y}_{i,j})$  from Eq. (3) for the data learnt by all tasks  $\{\tau_j | j = 1, \dots, k-1\}$ . We consider a maximum probability  $\sigma(\tilde{y}_{i,j})$  for defining an existing cluster. For a data sample  $\mathbf{x}_{k,i} \in \tau_k$ , we consider a new cluster, if after evaluating (3), we have :

$$\arg \max_{j=1}^{L_{K-1}} \sigma(\tilde{y}_{i,j}) < \Theta_2, \quad (7)$$

where  $\Theta_2$  defines new clusters. The memory management strategy associated with all clusters is explained in Appendix A from the Supplementary Material (SM).

## 5 Experimental Results

In this section, we evaluate the proposed unsupervised video continual learning methodology. We consider UCF101 [68], HMDB51 [19] and Something-Something V2 (SSv2) [11] datasets, after dropping all class labels, in order to use the data for unsupervised learning. Details about the datasets are provided in Appendix B from SM.

### 5.1 Implementation Details

We implement uVCL-KDE and uVCL-KDE-RBF using PyTorch [60] and the Adam optimizer [16] with a learning rate of 0.001, considering a single NVIDIA GeForce GTX 1080 Ti 11GB GPU. For each dataset, models are trained for up to 50 epochs, using a batch size of 8 videos. The model is optimized using the Focal Loss (FL) for balancing different video categories, as in Eq. (5) considering  $\gamma = 2$ , and using the Scikit-learn's framework [66] for the implementation. We utilize the Scaling Video Masked Autoencoders with Dual Masking (VideoMAE V2) [43] as the auto-encoder video transformer for implementing the feature extractor  $g(\cdot)$  to capture spatio-temporal features, as illustrated in Fig. 1. This transformer model is pre-trained on the Kinetics-700 dataset [15] without any label information. The input videos are composed of 16 frames, of size  $224 \times 224 \times 3$  pixels. During pre-processing, the video frames are center-cropped and their pixels re-scaled to the range



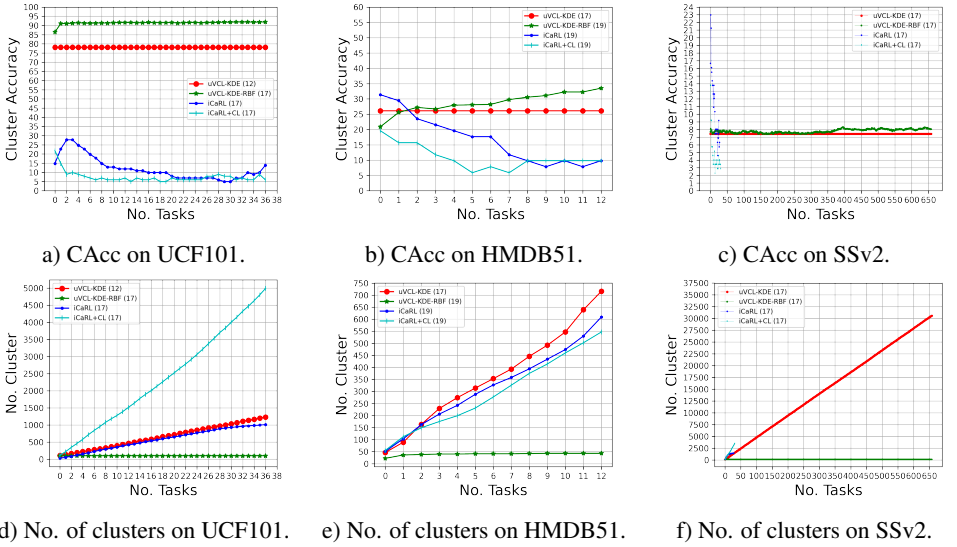


Figure 2: uVCL results on UCF101, HMDB51 and SSv2 considering the first fold data. Inside the brackets for each method we specify the bandwidth  $h$  for the mean-shift clustering.

[0.0, 1.0], then normalized using a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225]. The extracted output features for each video  $i$  at task  $k$  has  $|\mathbf{x}_{k,i}| = 1024$  channels. At the end of each task learning, we store the features for  $N = 20$  videos per each cluster in the memory buffers  $\mathcal{M}_i$ . The baselines used in the experiments are described in Appendix C from SM.

**Evaluation Metrics.** We adapt the protocol used in the unsupervised settings from [4, 40], evaluating the cluster accuracy (CAcc), used for the unsupervised continual learning for images [13]. Then, we evaluate the average unsupervised continual learning accuracy over all the training tasks, including the final task accuracy (ACAcc) as in [24, 44]. More details about the evaluation metrics used, including the Forward Forgetting (FWF) and the Backward Forgetting (BWF) are provided in Appendix D from SM. A large positive Forward Forgetting is also known as catastrophic forgetting.

## 5.2 uVCL-KDE results on UCF101, HMDB51 and SSv1

We apply the clustering methodology proposed in this paper in Section 4.2 for uVCL-KDE and also its extension uVCL-KDE-RBF, described in Section 4.3, where the linear units are randomly initialized. We present the results on UCF101, HMDB51, and Something-Something V2 (SSv2) datasets, where we split the data into 13, 37 and 659 tasks, considering 256 videos from a random mixed of classes, at each task. For UCF101 and HMDB51 datasets, the results averaged across three different data splits are shown in Table 1. In our experiments, we consider adjusting the bandwidth with various values for  $h$  for the Mean-shift algorithm in order to find the best value. We report the average final number of clusters ( $L_k$ ), average final cluster accuracy (CAcc) over three data splits for UCF101 and HMDB51, average cluster accuracy (ACAcc) over all learning tasks, backward forgetting (BWF), and forward forgetting (FWF). The best results are obtained for  $\Theta_2 = 0.3$  in Eq. (7), and for the



bandwidth  $h = 17$  for UCF101 and SSv2 whereas  $h = 19$  in HMDB51. According to the results from Table 1, uVCL-KDE-RBF achieves the best results for UCF101, HMDB51 and SSv2, by considering 100, 42 and 133 clusters, respectively.

Methods	h	UCF101					HMDB51					SSv2				
		Avg $L_k$	Avg CAcc <sub>k</sub>	Avg ACacc	BWF <sub>k</sub> ↑	FWF <sub>k</sub> ↓	Avg $L_k$	Avg CAcc <sub>k</sub>	Avg ACacc	BWF <sub>k</sub> ↑	FWF <sub>k</sub> ↓	Avg $L_k$	Avg CAcc <sub>k</sub>	Avg ACacc	BWF <sub>k</sub> ↑	FWF <sub>k</sub> ↓
uVCL-KDE	15	770	86.79	86.79	0.0	0.0	1,140	23.75	23.75	0.0	0.0	87,628	7.52	7.52	0.0	0.0
	16	657	86.90	86.90	0.0	0.0	914	26.51	26.51	0.0	0.0	59,895	7.68	7.68	0.0	0.0
	17	506	87.46	87.46	0.0	0.0	693	28.02	28.02	0.0	0.0	30,603	7.44	7.44	0.0	0.0
uVCL-KDE-RBF	18	518	82.98	82.98	0.0	0.0	486	27.26	27.26	0.0	0.0	15,658	7.27	7.27	0.0	0.0
	16	101	92.80	92.57	0.22	-0.11	117	22.45	21.34	1.08	0.23	227	7.42	7.46	0.01	0.000
	17	100	<b>93.45</b>	<b>93.01</b>	<b>0.33</b>	<b>-0.15</b>	92	27.79	25.97	1.48	0.08	133	<b>8.07</b>	7.81	0.24	-0.001
iCaRL [14]	18	93	88.27	88.05	0.16	-0.12	60	32.90	29.65	3.04	-0.67	79	8.02	7.64	<b>0.32</b>	-0.002
	19	85	83.52	83.31	0.25	-0.12	42	<b>34.05</b>	<b>29.81</b>	<b>3.76</b>	<b>-0.99</b>	67	7.44	7.36	0.10	-0.001
	17	727	10.23	12.34	-2.16	0.01	647	9.15	14.78	-6.10	1.58	1,486	6.32	<b>11.00</b>	-4.85	0.38
iCaRL+CL [15]	17	5,052	5.94	7.15	-1.25	0.27	542	11.11	10.76	0.38	0.38	3,553	3.45	4.92	-1.53	0.36
EWC [16]	42	2.31	1.92	0.40	-0.04	-0.04	13	3.92	3.82	0.11	-0.16	81	1.15	0.98	0.18	-0.02
MAS [17]	37	11.55	7.59	4.07	-0.29	-0.29	13	4.58	4.68	-0.11	-0.22	89	5.17	5.38	-0.22	<b>-0.16</b>

Table 1: Unsupervised video continual learning results for UCF101, HMDB51, and SSv2, where the results represent the average across three data splits. The results for the SSv2 dataset are provided only for the first 30 tasks.

We investigate the progressive learning of the proposed KDE-based methodology, task by task. The cluster accuracy (CAcc) are provided in Fig. 2-a, b, c, while the number of clusters considered according to increasing the number of tasks are provided in Fig. 2-d, e, f, respectively, for the continual learning of UCF101, HMDB51, and SSv2, respectively. These results show that uVCL-KDE-RBF achieves better results than all other baselines considered as well as than uVCL-KDE. The proposed method is shown to maintain and improve its performance over the successive learning of the tasks under all evaluation metrics. Moreover, our uVCL-KDE-RBF model finds a number of clusters which is close to the ground truth class number, assumed to be the number of classes. In addition, the baseline experiment on the SSv2 dataset is conducted only for the first 30 tasks because they require significant memory and significant computation costs for training, with the result showing a trend to a dramatic reduction in performance from the very beginning on this challenging dataset. Furthermore, results for the Backward Forgetting (BWF) and the Forward Forgetting (FWF) for all datasets are provided and explained in the Appendix E from SM.

### 5.3 Ablation study

**Changing the size of the memory buffer.** We consider storing the features corresponding to  $N = 20$  videos for each cluster, similar to the supervised video class incremental study from [14], where, unlike in our study, the class labels were known. Due to the inevitable variations in the size of each category, a small fixed memory size could lead to the loss of critical examples. To address this limitation, we consider a dynamic memory size for storing data, starting by keeping 10 examples per cluster and gradually expanding to 30 per cluster. The data stored is randomly selected from the data associated with each cluster. The results provided in Table 2 show that by increasing the memory size to 30 samples per cluster, results in a better performance on UCF101, while a smaller buffer of 10 examples is more effective for HMDB51.

**Changing the thresholds  $\Theta_1$  and  $\Theta_2$ .** For novelty detector threshold  $\Theta_1$  for UVCL-KDE in Eq (6), We use  $\Theta_1 = 16.53$ ,  $\Theta_1 = 16.92$ ,  $\Theta_1 = 15.98$ , for UCF101, HMDB51, and SSv2, respectively. Moreover, we vary the novelty detector threshold in Eq. (7), by increasing from  $\Theta_2$  which controls the confidence for creating new clusters and assigning them pseudo-labels

Methods	h	Memory Size	UCF101					HMDB51				
			Avg $L_k$	Avg $CAcc_k$	Avg $ACA_{Acc}$	$BWF_k \downarrow$	$FWF_k \uparrow$	Avg $L_k$	Avg $CAcc_k$	Avg $ACA_{Acc}$	$BWF_k \downarrow$	$FWF_k \uparrow$
uVCL-KDE-RBF	17	10 examples/	98	91.51	91.14	0.44	-0.16	87	27.44	25.86	<b>0.27</b>	<b>0.21</b>
	19	cluster	85	83.25	82.60	0.66	-0.17	46	<b>34.14</b>	<b>30.11</b>	3.82	-0.95
	17	30 examples/	97	<b>92.65</b>	<b>92.33</b>	0.31	-0.13	82	27.64	26.54	1.00	0.03
	19	cluster	86	84.39	84.20	<b>0.22</b>	<b>-0.08</b>	40	31.04	26.98	4.28	-0.72

Table 2: The performance on UCF101 and HMDB51 when we change the number of data stored in the memory buffers for each cluster, at 10 and 30 examples per cluster.

when learning new tasks. A smaller  $\Theta_2$  allows more clusters to be created, while higher thresholds are more selective, potentially avoiding incorrect cluster assignments. We experiment on UCF101 and HMDB51 datasets, and the results are shown in Table 3. We conclude that a small threshold at  $\Theta_2 = 0.3$  performs the best, resulting in semantically meaningful clusters. We provide the computation costs and the number of parameters required by each model in Appendix - F from the SM.

Methods	h	$\Theta_2$	UCF101					HMDB51				
			Avg $L_k$	Avg $CAcc_k$	Avg $ACA_{Acc}$	$BWF_k \uparrow$	$FWF_k \downarrow$	Avg $L_k$	Avg $CAcc_k$	Avg $ACA_{Acc}$	$BWF_k \uparrow$	$FWF_k \downarrow$
uVCL-KDE-RBF	17	0.7	118	<b>92.69</b>	<b>92.47</b>	<b>0.26</b>	-0.12	331	19.44	21.25	-1.54	0.75
		1.0	1,321	57.53	59.32	-1.78	0.88	748	10.99	13.99	-4.03	1.50
	19	0.7	106	85.72	85.51	0.23	<b>-0.18</b>	192	<b>28.10</b>	<b>28.29</b>	<b>-0.14</b>	<b>-0.55</b>
		1.0	1,180	59.54	60.43	-1.46	0.58	384	17.67	21.13	-3.89	0.30

Table 3: The performance on UCF101 and HMDB51 when changing the value of the threshold  $\Theta_2 \in \{0.7, 1.0\}$  for defining new clusters in Eq. (7). We store  $N = 20$  examples/cluster.

## 6 Conclusion and future work

In this paper, we propose a realistic yet effective framework for the Unsupervised Video Continual Learning (uVCL), which relies on dynamic kernel density estimation (KDE) representations for the features extracted by video transformers. A number of clusters is built and managed dynamically. We propose two different approaches, one based on the mean-shift algorithm for representing KDE and extracting clusters of video data while the other uses a linear layer on top of the clusters as in the Radial Basis Function (RBF) networks, and it is named uVCL-KDE-RBF. The key to sustaining the performance is to use memory buffers, storing the video features of some data associated with each cluster. Such stored data is then used again when new tasks are introduced, ensuring that the model can recall prior information and mitigate catastrophic forgetting. Our experiments highlight that our proposed methods not only that it reduces computation requirements and training time but it also effectively preserves past knowledge balancing stability and plasticity in the unsupervised video continual learning. In future work, we will employ a dynamic novelty detector criterion for deciding when to learn new information and define new clusters.

## References

- [1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory Aware Synapses: Learning what (not) to forget. In *Proc. of the European Conference on Computer Vision (ECCV)*, vol. LNCS 11207, pages 139–154, 2018.
- [2] Adrian G Bors and Moncef Gabbouj. Minimal topology for a radial basis functions neural network for pattern classification. *Digital Signal Processing*, 4(3):173–188, 1994.
- [3] Adrian G. Bors and Nikolaos Nasios. Kernel bandwidth estimation for nonparametric modeling. *IEEE Transactions on Sysmtes, Man, and Cybernetics- Part B Cybernetics*, 39(6):1543–1555, 2009.
- [4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proc. of the European conference on computer vision (ECCV)*, vol. LNCS 11218, pages 132–149, 2018.
- [5] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulò, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 9233–9242, 2020.
- [6] Chen Cheng, Jingkuan Song, Xiaosu Zhu, Junchen Zhu, Lianli Gao, and Hengtao Shen. Cucl: Codebook for unsupervised continual learning. In *Proc. of ACM International Conference on Multimedia*, pages 1729–1737, 2023.
- [7] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.
- [8] Dorin Comaniciu and Peter Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [9] Victor G Turrissi Da Costa, Giacomo Zara, Paolo Rota, Thiago Oliveira-Santos, Nicu Sebe, Vittorio Murino, and Elisa Ricci. Unsupervised domain adaptation for video transformers in action recognition. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1258–1265. IEEE, 2022.
- [10] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled outputs distillation for small-tasks incremental learning. In *Proc. European Conference on Computer Vision (ECCV)*. vol. LNCS 12365, pages 86–102, 2020.
- [11] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thureau, Ingo Bax, and Roland Memisevic. The “something something” video database for learning and evaluating visual common sense. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 5843–5851, 2017.

- [12] Xiaoshuai Hao, Wanqian Zhang, Dayan Wu, Fei Zhu, and Bo Li. Dual alignment unsupervised domain adaptation for video-text retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18962–18972, 2023.
- [13] Jiangpeng He and Fengqing Zhu. Unsupervised continual learning via pseudo labels. In *International Workshop on Continual Semi-Supervised Learning*, pages 15–32. Springer, 2021.
- [14] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13647–13657, 2019.
- [15] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.
- [16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proc. of the national academy of sciences (PNAS)*, 114(13):3521–3526, 2017.
- [18] Yajing Kong, Liu Liu, Maoying Qiao, Zhen Wang, and Dacheng Tao. Trust-region adaptive frequency for online continual learning. *International Journal of Computer Vision*, 131(7):1825–1839, 2023.
- [19] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 2556–2563, 2011.
- [20] Nattapong Kurpukdee and Adrian G. Bors. Temporal transformer encoder for video class incremental learning. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*, pages 1295–1301, 2024.
- [21] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 6028–6039. PMLR 119, 2020.
- [22] Hongbin Lin, Yifan Zhang, Zhen Qiu, Shuaicheng Niu, Chuang Gan, Yanxia Liu, and Mingkui Tan. Prototype-guided continual adaptation for class-incremental unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 351–368. Springer, 2022.
- [23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020.
- [24] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

- [25] Jiawei Ma, Xiaoyu Tao, Jianxing Ma, Xiaopeng Hong, and Yihong Gong. Class incremental learning for video action classification. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pages 504–508, 2021.
- [26] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. *arXiv preprint arXiv:2110.06976*, 2021.
- [27] Vali Ollah Maraghi, Karim Faez, et al. Class-incremental learning on video-based action recognition by distillation of various knowledge. *Computational Intelligence and Neuroscience*, 2022, 2022.
- [28] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2023.
- [29] Jaeyoo Park, Minsoo Kang, and Bohyung Han. Class-incremental learning for action recognition in videos. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13678–13687, 2021.
- [30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [32] Yixuan Pei, Zhiwu Qing, Jun Cen, Xiang Wang, Shiwei Zhang, Yaxiong Wang, Mingqian Tang, Nong Sang, and Xueming Qian. Learning a condensed frame for memory-efficient video class-incremental learning. In *Advances in Neural Infor. Proc. Systems (NeurIPS)*, pages 31002–31016, 2022.
- [33] Yixuan Pei, Zhiwu Qing, Shiwei Zhang, Xiang Wang, Yingya Zhang, Deli Zhao, and Xueming Qian. Space-time prompting for video class-incremental learning. In *Proc. of the IEEE/CVF International Conference on Computer Vision*, pages 11932–11942, 2023.
- [34] Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. *Advances in neural information processing systems*, 32, 2019.
- [35] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, 2017.
- [36] Arun Reddy, William Paul, Corban Rivera, Ketul Shah, Celso M de Melo, and Rama Chellappa. Unsupervised video domain adaptation with masked pre-training and collaborative self-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18919–18929, 2024.

- [37] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.
- [38] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [39] Abu Md Niamul Taufique, Chowdhury Sadman Jahan, and Andreas Savakis. Unsupervised continual learning for gradually varying domains. In *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3740–3750, 2022.
- [40] Wouter Van Gansbeke, Simon Vandenhende, Stamatis Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020.
- [41] Andrés Villa, Juan León Alcázar, Motasem Alfarrar, Kumail Alhamoud, Julio Hurtado, Fabian Caba Heilbron, Alvaro Soto, and Bernard Ghanem. PIVOT: prompting for video continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.
- [42] Andrés Villa, Kumail Alhamoud, Victor Escorcia, Fabian Caba Heilbron, Juan León Alcázar, and Bernard Ghanem. vCLIMB: A novel video class incremental learning benchmark. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recog. (CVPR)*, pages 19013–19022, 2022.
- [43] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560, 2023.
- [44] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 139–149, 2022.
- [45] Kun Wei, Xu Yang, Zhe Xu, and Cheng Deng. Class-incremental unsupervised domain adaptation via pseudo-label distillation. *IEEE Transactions on Image Processing*, 33 (1):1188–1198, 2024.
- [46] Tz-Ying Wu, Gurumurthy Swaminathan, Zhizhong Li, Avinash Ravichandran, Nuno Vasconcelos, Rahul Bhotika, and Stefano Soatto. Class-incremental learning with strong pre-trained models. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9591–9600, 2022.
- [47] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 374–382, 2019.
- [48] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

- [49] Fei Ye and Adrian G. Bors. Lifelong teacher-student network learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6280–6296, 2022.
- [50] Fei Ye and Adrian G. Bors. Lifelong generative adversarial autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):14684–14698, 2024.
- [51] Chengxu Zhuang, Tianwei She, Alex Andonian, Max Sobol Mark, and Daniel Yamins. Unsupervised learning from video with deep neural embeddings. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9563–9572, 2020.