Domain-Adaptive Online Active Learning for Real-Time Intelligent Video Analytics on Edge Devices

Michele Boldo*, Mirco De Marchi*, Enrico Martini*, Stefano Aldegheri and Nicola Bombieri

***Abstract*—Deep Learning (DL) for intelligent video analytics is increasingly pervasive in various application domains, ranging from Healthcare to Industry 5.0. A significant trend involves deploying DL models on edge devices with limited resources. Techniques such as pruning, quantization, and early-exit have demonstrated the feasibility of real-time inference at the edge by compressing and optimizing Deep Neural Networks (DNNs). However, adapting pre-trained models to new and dynamic scenarios remains a significant challenge. While solutions like domain adaptation, active learning, and teacher-student knowledge distillation contribute to addressing this challenge, they often rely on cloud or well-equipped computing platforms for fine tuning. In this study, we propose a framework for domain-adaptive online active learning of DNN models tailored for intelligent video analytics on resource-constrained devices. Our framework employs a knowledge distillation approach where both teacher and student models are deployed on the edge device. To determine when to retrain the student DNN model without ground-truth or cloud-based teacher inference, our model utilizes singular value decomposition of input data. It implements the identification of key data frames and efficient retraining of the student through the teacher execution at the edge, aiming to prevent model overfitting. We evaluate the framework through two case studies: human pose estimation and car object detection, both implemented on an NVIDIA Jetson NX device.**

***Index Terms*—Edge AI, Online Distillation, Edge Training, Human Pose Estimation, Real-time Training, Active Learning**

I. Introduction

Video stream analysis has emerged as a prominent use case for deep learning (DL), with notable applications from clinical gait analysis [1] to human-robot interaction in industrial environments [2]. Deep Neural Networks (DNNs) are particularly adapt at extracting spatial patterns from singular video frames. When applied to a sequence of streaming video, DNNs generate spatio-temporal patterns that are essential for tasks such as motion analysis, quality enhancement, object detection, and action recognition [3].

Deploying such models in real-world application scenarios poses significant challenges, particularly regarding their adaptability to specific contexts and addressing constraints related to real-time execution and data privacy. Two key issues arise when utilizing spatio-temporal models for video stream analysis. First, real-world conditions often differ substantially from the controlled environments in which pre-trained models were originally developed. These differences include variations of camera position, changes in lighting conditions, and quality of input data. Consequently, spatio-temporal models applied to streaming data necessitate fine-tuning to accommodate specific scenarios and environmental dynamics. Second, industrial and healthcare application environments often require real-time computing on edge devices to deal with latency, network bandwidth, and privacy constraints. The deployment of CNN-based deep learning models directly on resource-constrained edge devices [4] has gained considerable attention and various solutions have been proposed to achieve lightweight, efficient and accurate *inference* at the edge. Examples are model compression techniques such as neural network pruning [5], quantization [6], compact network design [7], as well as strategies like early-exit [8].

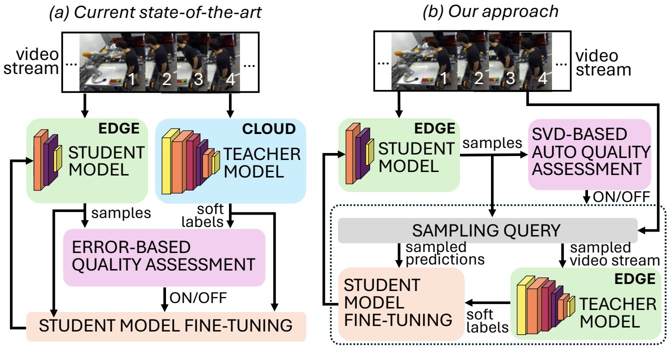


Fig. 1. Comparison between state-of-the-art approaches (a) and our proposed framework on domain-adaptive online active learning (b). In typical scenarios, the student and the teacher models are distributed in an edge-cloud cluster. Our approach leverages an auto quality assessment and active learning for full edge online fine-tuning.

However, integrating temporal space models deployment at the edge coupled with adaptation to new contexts presents additional challenges. This entails executing a training process to refine the model on a device with limited computational and memory resources. Since the training process is computationally heavy and resource-intensive, researchers have employed techniques to improve training efficiency [9], including the use of *active learning (AL)*. The goal of AL is to train a neural network only on the most significant samples to achieve high accuracy while minimizing training time. Nevertheless, as the environment undergoes continuous changes over time, contin-

This work has been supported by the "PREPARE" project (n. F/310130/05/X56 - CUP: B39J23001730005) - D.M. MiSE 31/12/2021, and the PNRR research activities of the consortium iNEST (Interconnected North-Est Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.5 –D.D. 1058 23/06/2022, ECS 00000043). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. Authors are with the Department of Engineering for Innovation Medicine, University of Verona, Italy - `name.surname@univr.it`. The authors thank Cesare Montresor for his assistance with the experimental results.

* equal contribution.

uous refinement of the model becomes essential. A potential solution is the *online domain adaptation* (ODA) paradigm [10], which aims at implementing continuous adaptation of the model to the application domain. One of the major issue of ODA is the unavailability of ground truth data and labels to fine-tune the model. A promising approach is the *knowledge distillation* (KD) [11], which relies on predictions made by a heavier and more accurate model (referred to as the *teacher*) to transfer knowledge to a light-weight model (referred to as the *student*).

However, deploying a KD architecture at the edge is computationally demanding due to the teacher overhead, particularly in scenarios with real-time constraints and limited resources.

Figure 1a illustrates the typical application of Online Domain Adaptation solutions in the current state of the art. Specifically, it involves the execution of the student model at the edge and the deployment of the teacher model in the cloud. When an error-based metric determines the necessity for training, images are transmitted to the cloud for label acquisition and subsequent training [12].

Additionally, employing offloading mechanisms to other edge or cloud devices is often unfeasible due to privacy constraints.

Efficiently scheduling the teacher model and training process to enable real-time student inference while also achieving domain adaptation remains an open challenge. One potential approach involves focusing training efforts on scenes where the model consistently makes errors. However, quantifying error is challenging, particularly when running the teacher model continuously is not feasible due to its impact on real-time system performance. To address these challenges, we present a framework that implements domain-adaptive online active learning of DNN models tailored for intelligent video analytics on resource-constrained devices (see Fig. 1b). The goal is to select optimal input sequences at run-time to train a lighter model with suboptimal accuracy, leveraging the predictions of a heavier, well-trained model as labels. Our framework implements an efficient auto quality assessment based on Singular Value Decomposition (SVD), by which it quantifies the accuracy of a student model in a sequence of predicted samples without relying on external references such as soft labels from a teacher or ground truth. By leveraging properties taken from the low-rank approximation theory, it enables efficient and automatic scheduling of the student model re-training.

It then employs the identification of pivotal data frames to implement efficient re-training of such a model through teacher execution at the edge, with the goal of mitigating model overfitting.

In summary, the novel contributions of this work are the following:

- An SVD-based auto quality assessment that determines the need for student model fine-tuning without any ground truth or teacher's labels;
- An online active learning (OAL) framework that implements an online domain adaptation at the edge through student-teacher knowledge distillation retraining in real-time;
- An analysis of different algorithms to select the appropriate video frames aimed at facilitating model adaptation while mitigating the risk of overfitting;

We present an evaluation of the proposed framework through two case studies: human pose estimation and automotive object detection, implemented using standard and widely used platforms (TRTPose [13], OpenPose [14], YOLO [15], SSD [16]) employing DNN architectures such as ResNet [17], DenseNet [18], and Darknet [19] on an NVIDIA Jetson NX edge device.

Our evaluation encompasses both standard datasets (COCO [20], H3.6M [21], VOC [22], LaSOT [23]) and a real-world scenario (i.e., an HPE dataset collected on a smart manufacturing line) for quantitative assessment. Furthermore, we compare our approach with state-of-the-art methodologies to underscore its effectiveness, particularly in minimizing unnecessary training in static scenarios, resulting in up to a 90.9% reduction in domain adaptation efforts.

II. Background and related work

Active learning (AL) has been proposed as a dynamic training strategy that involves the active selection of a reduced set of data samples [24]. Lu et al. [25] proposed a performance benchmark of AL strategies for binary classification. Liu et al. [26] proposed an Active Learning (AL) framework to reduce the amount of annotations required in a large unlabelled dataset. They employed a sampling strategy based on uncertainty, training the network only on frames where the Human Pose Estimation (HPE) model is not confident. Yoo et al. [27] developed a loss prediction model that learns how to limit the loss defined in the target model. Zhang et al. [28] proposed a query strategy that selects frames balancing between information representativeness and uncertainty.

Out-of-distribution (OOD) detection in machine learning involves identifying instances in test data that significantly differ from the training data. In [29] the authors propose a unified framework that uses *energy* as a cost function to determine if a sample is out-of-distribution. Additionally, they propose a trainable cost function to enhance the classification model. This approach in active learning can enhance model robustness by preventing confident predictions on unfamiliar data and facilitate dataset expansion by flagging novel samples. OOD detection can also improve uncertainty estimation and optimize resource allocation for labelling efforts. In [30], the authors present a methodology in which the human pose is modelled using a Bayesian network trained through maximum likelihood estimation. Poses with low likelihood are identified as out-of-distribution, making them excellent candidates for annotation.

Online Active Learning (OAL) is an extension of traditional Active Learning, designed for data that becomes available incrementally. This approach efficiently utilizes resources by selecting a subset of the data while maintaining accuracy even when the data distribution changes, making it valuable in scenarios that require rapid and timely responses. Cacciarelli et al. [31] presented an overview of recent methods for selecting informative observations from streaming data. Manjah et al. [32] introduced a real-time OAL framework aimed at fine-tuning a lightweight model for object detection in videos. Nevertheless, none of these methods evaluate the effectiveness and necessity of training, they focus only on selecting the most informative input samples.

Mullapudi et al. [12] presented ”Just-in-Time”, an OAL framework that employs model distillation to create efficient, low-cost semantic segmentation models for specific video streams. They achieve significant runtime cost reductions without offline pre-training. In particular, the algorithm decides when to train the student model using an accuracy-based metric with respect to the teacher. Khani et al. [33] proposed Adaptive Model Streaming, an edge-cloud method to boost lightweight model performance for real-time video inference. It adapts compact models through online knowledge distillation in video semantic segmentation, minimising the bandwidth usage. They execute both teacher and training on a cloud server. To ensure that the transmission network load is manageable, they limit the sent samples by selecting only a subset of the collected frames. Training is initiated after accumulating a predefined number of samples, and the frame sampling rate varies according to the vehicle speed in an automotive context. Their approach is claimed to outperform the one presented in [12] reducing also the model overfitting.

Traditional CNN training methods frequently have limitations in the context of edge computing, where computational resources are limited. In addition, these approaches depend on the continuous execution of the teacher model to regularly assess the accuracy of the student model. In resource-constrained platforms, it is not always feasible to rely on the execution of the teacher. Our platform executes an OAL framework at the edge, thus avoiding the teacher inference. The platform is equipped with a smart SVD-based auto quality assessment specifically designed for *spatio-temporal neural network*.

Spatio-temporal models [34] represent a distinct class of models designed to process spatio-temporal data, such as those used in human pose estimation and object detection. Spatio-temporal data is a sequential structure that captures spatial relationships over time. For example, in the analysis of human motion from video streams, a human pose estimation system generates a sequence of human poses over time. This sequence exhibits temporal coherence, allowing the evaluation of spatial correlations between poses at consecutive time points. Similarly, these principles are applicable to object detection tasks. Spatio-temporal models exhibit two fundamental properties: (*A*), the output signal is related to the input space, as a coordinate in a 2D or 3D reference system; (*B*), the signal shows temporal correlation, meaning that the input and output at time t are closely related to the signal at time $t-1$.

Singular Value Decomposition (SVD) is a fundamental concept in linear algebra that decomposes a matrix into three simpler matrices. SVD is widely used in various fields, including video processing, due to its ability to reveal underlying patterns and structures in data [35]. One common application in video processing is low-rank approximation for image compression [36]. A video is represented as a series of frames stacked together into a matrix. These frames often exhibit redundancy and correlation, especially in the spatial and temporal dimensions. Low-rank approximation using SVD exploits this redundancy by approximating the original video matrix with a lower-rank approximation, thus compressing the video data while retaining essential information. Since SVD captures both spatial and temporal correlations in input matrix, it synergizes effectively with spatio-temporal models. SVD is especially useful for tasks such as denoising, compression, and feature extraction in video processing, where capturing both spatial and temporal correlations is crucial for accurate analysis and efficient data representation [3].

In this work, we propose a methodology to identify instances when the neural network makes errors in a spatiotemporal task, based on the aforementioned concepts. Our method operates independently of the network's confidence, which solely considers the network's probability output and does not account for actual spatiotemporal errors such as jitter. Instead, our framework focuses on the quality of the final prediction. This methodology incorporates active learning to reduce the number of samples required during training, making our framework suitable for online fine-tuning at the edge.

III. Methodology

Figure 2 shows an overview of the OAL platform based on SVD auto quality assessment. The platform implements OAL through two distinct models: a lightweight model (i.e., the student) and a more accurate yet computationally intensive model (i.e. the teacher). Due to the inherent characteristics of the SVD technique used, our framework targets spatio-temporal models. Therefore, both the teacher and student models must exhibit the space-time properties *A* and *B* described earlier. To demonstrate the methodology's generalizability, we evaluated it on two spatio-temporal tasks: human pose estimation and object detection. Since the platform is designed to be deployed on edge devices, the aim is to minimise the computational load by only activating re-training when necessary. Furthermore, this approach allows the system to minimise latencies, thus ensuring a higher quality of service.

The student processes a video stream taken from an RGB sensor, generating a series of samples. A generated sample S_i is a set of J keypoints in the two-dimensional image space representing the location of a target class predicted by the student (e.g. a body joint for human pose estimation, a vertex of a bounding box for object detection). The framework maintains two fixed-length queues, one for the student samples and the other for the input video frames. The SVD-based auto quality assessment determines the quality of the student's samples and evaluates the need to activate the OAL platform. When the SVD-based auto quality assessment requests a student model's fine-tuning, then the teacher generates the soft-labels for retraining. To reduce computation time and overload on teacher and student train executions, the OAL platform applies a sampling query on the video queue and the sample queues.

In changing and dynamic scenarios, the student's samples extrapolated by the inference phase are imprecise, as the initial training dataset and the working environment may be different. The only way to improve the accuracy of the predictions is through fine-tuning. Since ground-truth labels are not available in real-world scenarios, we use the knowledge distillation paradigm, by which the teacher provides the *soft labels*. In [12], the authors periodically employ the teacher model on single frames, comparing its outcomes with those of the student model. The training begins whenever the accuracy exceeds a certain threshold, and it is suspended otherwise. This requires the training images to be processed also by the teacher, thus decreasing the performance of the edge device.

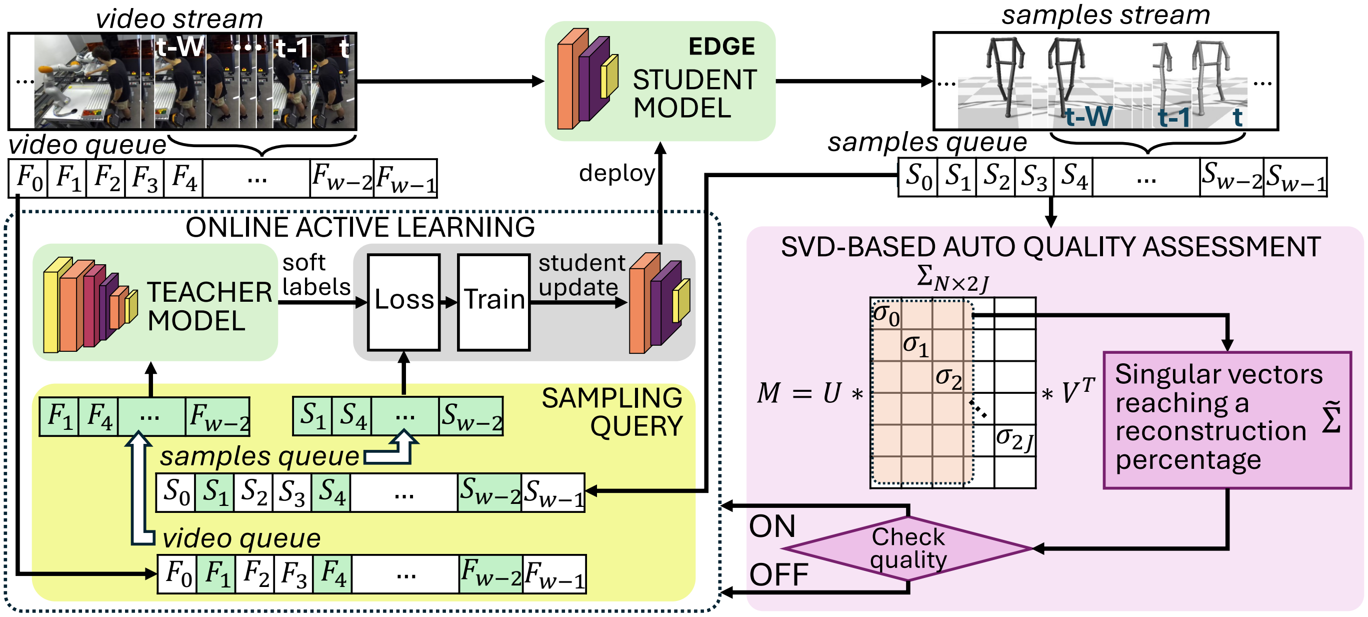

video stream
... t-W ... t-1 t
video queue
F_0 F_1 F_2 F_3 F_4 ... F_{w-2} F_{w-1}
EDGE
STUDENT MODEL
deploy
samples stream
... t-W t-1 t ...
samples queue
S_0 S_1 S_2 S_3 S_4 ... S_{w-2} S_{w-1}
ONLINE ACTIVE LEARNING
TEACHER MODEL
soft labels
Loss
Train
student update
SAMPLING QUERY
F_1 F_4 ... F_{w-2}
S_1 S_4 ... S_{w-2}
samples queue
S_0 S_1 S_2 S_3 S_4 ... S_{w-2} S_{w-1}
video queue
F_0 F_1 F_2 F_3 F_4 ... F_{w-2} F_{w-1}
SVD-BASED AUTO QUALITY ASSESSMENT
$\Sigma_{N\times 2J}$
$M = U *$
σ_0
σ_1
σ_2
σ_{2J}
$* V^T$
Singular vectors reaching a reconstruction percentage $\tilde{\Sigma}$
Check quality
ON
OFF

Fig. 2. The proposed online active learning platform relying an auto quality assessment based on singular value decomposition for student video stream processing and teacher-student model knowledge distillation with sampling query optimization.

Our platform, in contrast, relies only on information provided by the student to determine when training should start. This approach activates the teacher only when acquiring soft labels is necessary, thereby preventing a slowdown in the system's throughput during the prediction quality assessment phase.

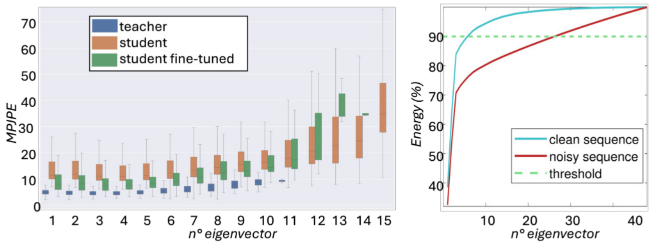
We define $\mathcal{V}$ as the space of video frames, where each frame $F \in \mathcal{V}$ is a two-dimensional array of pixels.

We represent each frame F as a spatial grid of pixels $p_{i,j}$, where $F = [p_{i,j}]$ for $i = 1, 2, \dots, M$ and $j = 1, 2, \dots, N$, with M and N denoting the height and width of the frame, respectively. The spatial representation captures the static arrangement of objects within individual frames, encoding information about color, texture, and spatial patterns. The temporal dimension represents the sequence of video frames over time.

As shown in Figure 2, we define the video queue $VQ = \{F_0, F_1, F_2, \dots, F_{W-1}\}$ as the queue of video frames, where W represents the total number of frames in the video sequence. Each frame F_t is indexed by t, representing the time index. Finally, we define the samples queue $SQ = \{S_0, S_1, S_2, \dots, S_{W-1}\}$ as the corresponding samples (i.e. the student's predictions) of the video frames.

A. SVD-based Auto Quality Assessment

To choose when to perform training, we apply principles from the low-rank approximation theory to a samples queue along with their corresponding video frames in the video queue. The objective is to determine how many *singular vectors* are required to accurately reconstruct the sample sequence. We represent the sequence of samples as a matrix M with n rows and $2J$ columns, where n denotes the number of collected frames and J represents the number of 2D key points for each sample. Given the inherent connectivity of body segments through physical articulations and the extensive use of motion synergies by the motor cortex in human motion [37]–[39], it is possible to efficiently represent sample sequences in lower dimensions. Consequently, samples exhibit high correlation and can be approximated effectively using low-rank matrices [40]. The temporal correlation captures repetitive movement patterns, while the spatial correlation reflects the kinematic synergies of the human body. The matrix M can be decomposed into three sub-matrices using the SVD method [41], as expressed in Equation (1):



(a) (b)

Fig. 3. Cumulative sum of the values of Σ from a clean and a noisy sequence.

$$M = U\Sigma V^T \tag{1}$$

where U and V are orthonormal matrices of dimensions $\mathbb{R}^{n\times n}$ and $\mathbb{R}^{2J\times 2J}$, respectively, and they contain information about the spatial and temporal properties among the samples. The diagonal weight rectangular matrix Σ in $\mathbb{R}^{n\times 2J}$ contains singular values sorted by importance, where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{2J} \in \Sigma$. SVD offers a hierarchical representation of human motion data based on dominant correlations. By truncating the singular values and their corresponding singular vectors, we can approximate the history matrix with a reduced rank. Truncation acts on the diagonal weight matrix Σ and allows the approximated matrix $\tilde{M}$ as defined in Equation (2):

$$\tilde{\Sigma} = [\sigma_1\sigma_2\dots\sigma_r\;0\dots0] \quad \tilde{M} = U\tilde{\Sigma}V^T \quad r < 2J \tag{2}$$

where the threshold rank r splits the main motion trends and noise to obtain the approximated matrix $\tilde{M}$.

According to the Eckart-Young Theorem [42], the best possible low-rank approximation of matrix M obtained by minimizing the Frobenius norm of the difference between M and the approximated matrix $\tilde{M}$ is achieved through the truncated SVD of M.

In cases where the data is well-structured and with strong inter-dependencies, a significant portion of the information is concentrated in the first singular vectors. In the presence of noise, the absence of coherent patterns and the presence of random fluctuations lead to reduced correlations among data points. Consequently, the information becomes more evenly distributed across the singular values spectrum. Figure 3a illustrates the correlation between the Mean Per Joint Position Error (MPJPE) and the number of eigenvectors required to reconstruct 98% of the original information. Specifically, it analyzes three modalities: a highly accurate teacher, a less accurate student, and the same student after retraining. When only one eigenvector is needed to achieve 98% of the information, the MPJPE is very low (approximately 10 pixels). As the number of eigenvectors increases, the MPJPE correspondingly increases. It is also noteworthy that both the teacher and the student, after fine-tuning, always require fewer than 14 eigenvectors to reconstruct 98% of the information, even in the noisiest tracks. Figure 3b shows two normalized cumulative sums of the diagonal matrices Σ extracted from the sample sequence. These are derived from two scenarios: with and without noise. The first 6 singular vectors capture over 90% of the information in the clean sequence, whereas it takes 26 singular vectors from the noisy sequence to retain the same level of information.

It is important to note that this methodology identifies noisy data independently of the underlying cause. Errors due to changes in lighting, background, and Out of Distribution (OOD) data are detected. The latter is particularly interesting as OOD for human pose estimation may arise from "difficult" poses that the student model has not encountered during training. These challenging poses are detected by the reduced accuracy of the student model, which is subsequently identified using our SVD-based methodology. However, these poses may not be OOD for the teacher model, which then predicts soft labels to train the student model. According to this principle, we developed a methodology to extrapolate these values through a microbenchmarking phase. Algorithm 2 outlines the computational flow, given a dataset D, a teacher model T_m, a student model S_m and a predefined time window W. To improve the real-world applicability of the proposed methodology, D can be a sequence of images collected by the end user in various application scenarios, as there is no requirement for ground truth data. In this phase, we assume the teacher model is sufficiently accurate to predict a clean sequence. In detail, the framework first utilizes the procedure outlined in 1 to extrapolate the average information percentage of each eigenvector for both the clean (teacher) and noisy (student) sequences. This procedure performs an inference on all D, extracting the predictions for the considered $Model$. Subsequently, the algorithm computes a normalized cumulative sum c for each eigenvector to obtain the information percentage within a predefined sample window W through an iterative process. Once these values are obtained for each time window and stored in C (a matrix with $\frac{N}{W}$ rows and $2J$ columns), the algorithm calculates the mean information percentage for each eigenvector across the windows (rows). The number of eigenvectors τ is determined by identifying the eigenvector that, on average, shows the greatest percentage difference in energy between the clean and noisy sequences. The percentage of information threshold value θ is set based on the average energy of the noisy sequence in the selected eigenvector.

Algorithm 1 Reconstruction percentage for each eigenvector

```
Inputs: $D$, $Model$, $W$
Output: $\tau, \theta$
Initialize: $N \leftarrow |D|$, $C \leftarrow \emptyset$
$predictions \leftarrow$ Predict($Model, D$)
for $start = 0$ to $N - W + 1$ step $W$ do
    $end \leftarrow start + W$
    $W \leftarrow predictions[start : end]$
    $U, \Sigma, V \leftarrow SVD(W)$
    $c \leftarrow \left\{ \frac{\sum_{i=0}^{\tau} \sigma_i}{\sum_{i=0}^{2J} \sigma_i}, \ \forall \tau \in [0, 2J] \right\}$
    $C \leftarrow C \cup \{c\}$
end for
$E \leftarrow$ Mean of $C$ for each eigenvector (along rows)
```

Algorithm 2 Analysis of Model Predictions

```
Inputs: $D$, $T_m$, $S_m$, $W$
Output: $\tau, \theta$

$E\_T \leftarrow$ Algorithm 1$(D, T\_m, W)$
$E\_S \leftarrow$ Algorithm 1$(D, S\_m, W)$
$E \leftarrow E\_T - E\_S$
$\tau \leftarrow argmax(E)$
$\theta \leftarrow E\_S[\tau]$
```

The reconstruction percentage is a normalized cumulative sum of the eigen values on $\tilde{\Sigma}$ diagonal weights in $[0, \tau]$ range. When the value of the percentage doesn't reach a threshold of reconstruction θ, then the sequence is considered noisy, as expressed in Equation (3):

$$\frac{\sum\limits_{i=0}^{\tau} \sigma_i}{\sum\limits_{i=0}^{2J} \sigma_i} < \theta \tag{3}$$

The cumulative sum of the $\tilde{\Sigma}$ matrix is normalized by the term in the denominator. The training process starts if the normalized cumulative sum of eigen values doesn't exceed a defined reconstruction percentage (i.e., the sequence is considered noisy).

Algorithm 3 provides an overview of the proposed OAL procedure. It relies on two user-defined input variables: τ and θ, where τ represents the maximum number of singular vectors used in the SVD approximation to achieve the target information percentage θ. The algorithm uses three memory buffers: $batch$ to store the images selected for training with a sampling strategy $\mathcal{S}_p$, $labels$ to store the soft labels generated by the teacher, and kps_s as the sequence of samples predicted by the student. At time i, the student processes the image I_i with the model weight configuration Ω_i, yielding the predicted sample kps^s_i, which is added to the sequence kps^s. Lines 6-8 highlight the training scheduler role in performing the SVD and determining whether fine-tuning of the student model is necessary. After training (line 10), the procedure optimizes the new weights, through pruning and quantization, before deploying to the student.

Algorithm 3 Online Active Learning Framework

```
    Inputs: $\tau$, $\theta$, $\mathcal{S}_p$
    Output: $\text{kps}^s_i$
    Initialize: batch $\leftarrow \emptyset$, labels $\leftarrow \emptyset$, $\text{kps}_s \leftarrow \emptyset$
 1: for $i \leftarrow 0$ to $n$ do
 2:     $\text{kps}^s_i = \text{student}(\Omega^s_i, I_i)$
 3:     $\text{kps}_s \leftarrow \text{kps}_s \cup \{\text{kps}^s_i\}$
 4:     batch $\leftarrow \mathcal{S}_p(\text{kps}_s)$
 5:     if $i \mod \mathit{\Delta} \equiv 0$ then
 6:         $U, \Sigma, V = \text{SVD}(\text{kps}_s)$
 7:         if Equation3$(\Sigma)$ then
 8:             labels $= \text{teacher}(\Omega^T, \text{batch})$
 9:             $\Omega^s_{i+1} = \text{training}(\Omega^s_i, \text{batch}, \text{labels})$
10:             $\Omega^s_{i+1} = \text{optimization}(\Omega^s_{i+1})$
11:         else
12:             batch $\leftarrow \emptyset$
13:             $\text{kps}_s \leftarrow \emptyset$
14:         end if
15:     end if
16: end for
```

It is important to note that the system aims to guarantee the inference process in real-time, ensuring quality of service and continuous prediction of human poses. The approach detects the optimal times to conduct the training, which does not need to be real-time, as for example time windows when there are no people in the scene.

B. Sampling Query

To iteratively select the most informative data samples for teacher annotation and training, we explore four sampling metrics used for fine-tuning: uniform, random, confidence and error sampling. To define a sampling query strategy, we need the indexes of the video queue VQ and the samples queue SQ. Let I be the set of common indexes (e.g. $I = \{t, t+1, \ldots, t+W-1\}$), and W is the length of the queues. A sampling strategy is a function that selects a subset $I' \subseteq I$ based on a specific criterion or rule. Formally, a sampling function is $\mathcal{S} : I \rightarrow \mathbb{B}$, where I is the set of common indexes and $\mathbb{B}$ is the boolean domain such that $\mathbb{B} = \{true, false\}$. The sampling function $\mathcal{S}$ allows to extract a mask from the original domain I to obtain the sampled subset $I' \subseteq I$ as follow: $I' = \{i \in I : \mathcal{S}(i)\}$. From the definition of a sampling function on single indices (I), we expand the definition to a set of indices ($Set[I]$), since in general a sampling strategy applies a sorting on the entire set of input samples based on a specific criterion. We define a sampling strategy as $strategy\_sampling : Set[I] \rightarrow Mask[\mathbb{B}]$, where $Set[I]$ is the input set of indexes in I and $Mask[\mathbb{B}]$ is the output boolean mask.

The sampling metrics employed in this study are implemented as fixed-rate strategies, where they operate by selecting a specific percentage of frames within a defined window of samples. Given a window of indexes $w \in Set[I]$ and a percentage $p \in P$, a fixed-rate sampling metric is defined as $fixed\_sampling : (Set[I], P) \rightarrow Mask[\mathbb{B}]$, where I is the set of queue indexes as previously defined and $P = \mathbb{R}_{[0,1]}$.

Given $b = fixed\_sampling(w, p)$ a boolean mask from a fixed sampling of window indexes w and percentage p, the following are the properties of a fixed-rate sampling: (1) $\{i \in w : b[i]\} \subseteq w$; (2) $|\{i \in w : b[i]\}| = |w| * p$; (3) $fixed\_sampling(w, 1) \equiv [true, true, \ldots, true]_{\mathbb{B}}$ and $fixed\_sampling(w, 0) \equiv [false, false, \ldots, false]_{\mathbb{B}}$. The properties 1 and 2 establish that fixed-sampling involves extracting a subset of the original input set of samples, where the size of the resulting set is determined by a percentage that corresponds to the specified proportion of the initial set's cardinality. Finally, property 3 emphasises that if you apply sampling at percentage 1, then you get no-sampling, with percentage 0 instead you get empty set.

The implemented sampling metrics in this study all utilize fixed-rate sampling. The following will provide definitions for each of the sampling metrics: uniform sampling, random sampling, confidence sampling, and error sampling.

Uniform sampling is a straightforward approach where data samples are selected uniformly from the window with a specific rate that is inferred from the input percentage: $rate = 1/p$. It does not take into account any specific criteria or model information and treats all unlabeled samples equally. Similarly to the $fixed\_sampling(w, p)$ function, the uniform sampling is defined as a function that takes in input a set of samples w and the percentage p to extrapolate and gives $uniform\_sampling(w, p) = [i \bmod 1/p = 0 : i \in w]$. Similar to uniform sampling, random sampling selects data samples randomly from the dataset. While uniform sampling may utilize a deterministic technique, random sampling can use more sophisticated randomization methods, such as stratified random sampling, to ensure a more representative selection of samples. As previously, the random sampling is defined as a function that takes in input the window indexes w and the percentage p and gives $random\_sampling(w, p) = [random() > p, \forall i \in w]$, where $random(n)$ is a function, defined as $random : \cdot \rightarrow \mathbb{R}_{[0,1]}$, that randomly return a number from 0 to 1. Confidence sampling aims to select data samples based on the model's confidence in its predictions. For DNN models, this metric calculates the uncertainty or confidence score associated with each sample. Samples with high uncertainty or low confidence are more informative and have higher priority for annotation, as they are likely to be challenging or ambiguous cases for the model. The confidence sampling is defined as a function that takes in input the window indexes w and the percentage p and gives $confidence\_sampling(w, p) = \{c_i > \lfloor W * p \rfloor : c_i \in argsort(confidence(SQ[w]))\}$, where $confidence(samples)$ is a function, defined as $confidence : Set[SQ] \rightarrow Set[\mathbb{R}_{[0,1]}]$, that gives the confidence of each samples in input of the samples queue SQ, and

$argsort(confidences)$ is a function, defined as $argsort : Set[\mathbb{R}] \rightarrow Set[I]$ that sorts the input set of confidences in ascending order, returning each sort index. The confidence sampling provides the first $\lfloor W * p \rfloor$ samples with lower confidence from the input samples in $SQ[w]$. Error-based sampling is a sampling metric that selects frames with the highest error in comparison to a ground truth. It exploits the ground truth annotations to identify the frames where the model's predictions deviate the most from the true pose. By selecting these challenging frames, error-based sampling aims to provide the model with the most erroneous predictions for improving its performance. The error-based sampling is defined as a function that takes in input the window indexes w and the percentage p and gives $error\_sampling(w, p) = \{e_i > \lfloor W * p \rfloor : e_i \in argsort\_descend(error(SQ[w], GT[w]))\}$. The $error(samples, ground\_truth)$ is a function, defined as $error : (Set[SQ], Set[GT]) \rightarrow Set[\mathbb{R}^+$, that calculates the distance from a set of ground-truth GT of each corresponding sample of the samples queue SQ, and $argsort\_descend(errors)$ is a function, defined as $argsort\_descend : Set[\mathbb{R}] \rightarrow Set[I]$ that sorts the input set of confidences in descending order, returning each sort index. The error sampling provides the first $\lfloor W * p \rfloor$ samples with higher distance (i.e. error) between the input samples $SQ[w]$ and the ground truth $GT[w]$.

IV. Experiments

We evaluated the proposed framework on two types of spatio-temporal models: Human Pose Estimation (HPE) and Car Object Detection (OD). HPE is a computer vision task which consists of extrapolating 3D human body keypoints from images or videos. It employs convolutional neural networks (CNNs) and platforms such as TRTPose [13] and OpenPose [14]. Recent advancements have significantly improved the accuracy of HPE systems to deal with limitations like occlusions, varying lighting conditions, and complex or unusual body poses. We first adopted a standard dataset to quantitatively assess the accuracy of the proposed platform with an infra-red marker-based system as ground truth. We also adopted a real-world dataset, which consists of videos of collaborative human-robot interaction in a Industry 4.0 scenario. Furthermore, we evaluated different query strategies and training samples' percentage for active learning in offline and online HPE scenario.

OD is a computer vision process that identifies and localizes multiple objects within an image or video frame. It involves detecting the presence, location, and often the class of objects within the visual data. We adopted state of the art models for OD, such as YOLO [15] and SSD [16], which provide high speed and accuracy for real-time detection tasks.

The Algorithm 3 input parameters, τ, θ and p, defined in Section III, were fine-tuned with a preliminary study. The resulting values are determined by running through microbenchmarking. Although we kept those values constant throughout the experiments, other datasets may need specific parameters. We evaluated both platforms on an NVIDIA Jetson NX equipped with a 384 CUDA cores GPU accelerator, 8 GB of unified memory and a 6-core processor.

A. Results on Human Pose Estimation

The student model consists of the TRTPose HPE platform [13], which employs a *Densenet* CNN pre-trained on the *COCO-Pose* dataset [20]. Because of the substantial differences between this dataset and those used for our quantitative and qualitative analyses, the performance of this model is suboptimal. The teacher model is OpenPose [14], utilizing a BODY25 CNN pre-trained on the *MPII* and COCO-Pose datasets. This model demonstrates excellent generalization capability, performing effectively in most scenarios. For the HPE case study, using Algorithm 2, we extrapolated the following input parameters for Algorithm 3: $r = 14, \theta = 99\%, sampling = 5\%$.

1) Auto quality assessment: For the quantitative analysis, we tested our framework on Human3.6M [21], a widespread standard dataset for HPE. It consists of several sets of human actions and movements recorded in controlled indoor environments. It includes actions such as walking, running, sitting, standing, greetings, eating, smoking, waiting, discussions, and posing. The ground truth is a 3D motion capture data synchronised with the corresponding 2D video frames. To test the framework in the OAL context, we concatenated the actions of a subject (S1) to obtain a single long video sequence. We created three scenarios:

- *Dynamic*: in this scenario, we evaluate the framework performance handling diverse and unpredictable scenes. To do this, we concatenated all scenes in lexicographic order.
- *Semi-repetitive*: it represents a structured environment, in which a static and a dynamic scene follow each other in a repeated pattern.
- *Repetitive:* it recreates a typical scene in an industrial environment (e.g., in production line), where a person remains in similar positions frequently. We concatenated all similar scenes, such as the act of having a discussion and making gestures.

We present the comparison of our methodology for on-demand training with two other schedulers. The first (*train always*) performs training every t seconds. This methodology, from the perspective of model adaptability, should be considered the best reference model. Training the student model on the entire sequence incrementally is expected to yield the best results. However, this approach is not feasible at runtime, as the student model would be continuously trained, introducing significant latencies. The second (*accuracy-based*) represents the current state-of-the-art approach [12], which triggers the training phase when the average error of previous instances exceeds a certain threshold.

Table I shows the accuracy and performance results of the different training schedulers. The evaluation metrics include *Mean Per Joint Position Error* (MPJPE), the update percentage (i.e., the amount of student training), and the percentage of teacher executions to obtain soft labels and reference results. The dynamic scenario is the most challenging due to its high variability, requiring continuous domain adaptation. The proposed solution performs well in this dynamic setting, achieving performance similar to the accuracy-based approaches. On the other hand, the semi-repetitive and repetitive scenarios demands less domain adaptation. In this context, our solution achieves up to 90.6% reduction in the number of training

TABLE I
ACCURACY OF THE OAL PLATFORM IN HUMAN POSE ESTIMATION TASK (HPE) VARYING THE TRAINING SCHEDULES.

Sequence	**Dynamic scenario**			**Semi-repetitive scenario**			**Repetitive scenario**		
Metric	**MPJPE**	**%update**	**%teacher execution**	**MPJPE**	**%update**	**%teacher execution**	**MPJPE**	**%update**	**%teacher execution**
Teacher	7.9	-	-	7.2	-	-	7.6	-	-
No train	13.2	0.0	0.0	10.7	0.0	0.0	13.2	0.0	0.0
Train always	11.7	100.0	100.0	8.4	100.0	100.0	9.5	100.0	100.0
Accuracy-based	12.1	69.7	100.0	9.0	12.1	100.0	9.5	3.0	100.0
Our	12.2	74.2	74.2	8.5	60.6	60.6	9.4	9.1	9.1

TABLE II
ENERGY CONSUMPTION AND LATENCIES OF THE OAL PLATFORM VARYING THE TRAINING SCHEDULES IN HPE TASK ON EDGE DEVICE AND SERVER.

Device	**NVIDIA Jetson Xavier NX**									**NVIDIA GeForce RTX 2070 SUPER**								
Sequence	**Dynamic**			**Semi-repetitive**			**Repetitive**			**Dynamic**			**Semi-repetitive**			**Repetitive**		
Performance	**Energy**	**Time**		**Energy**	**Time**		**Energy**	**Time**		**Energy**	**Time**		**Energy**	**Time**		**Energy**	**Time**	
	mWh	**ms**	**Hz**	**mWh**	**ms**	**Hz**	**mWh**	**ms**	**Hz**	**mWh**	**ms**	**Hz**	**mWh**	**ms**	**Hz**	**mWh**	**ms**	**Hz**
Teacher	31.2	168.1	5.9	31.2	168.1	5.9	31.2	168.1	5.9	175.1	72.2	13.8	175.1	72.2	13.8	175.1	72.2	13.8
Student	18.2	30.8	32.4	18.2	30.8	32.4	18.2	30.8	32.4	100.3	7.2	139.7	100.3	7.2	139.7	100.3	7.2	139.7
Student while teacher	33.5	49.4	20.2	33.5	49.4	20.2	33.5	49.4	20.2	186.6	8.7	114.7	186.6	8.7	114.7	186.6	8.7	114.7
Student while training	27.9	56.0	17.9	27.9	56.0	17.9	27.9	56.0	17.9	168.2	10.6	94.6	168.2	10.6	94.6	168.2	10.6	94.6
Accuracy-based	31.7	51.41	19.4	28.6	55.2	18.1	28.1	55.8	18.0	173.5	9.9	100.0	170.2	10.4	96.7	168.5	10.5	95.2
Our	27.5	47.1	21.3	25.8	44.1	22.7	19.3	32.8	30.5	157.1	8.9	111.2	109.3	7.5	134.2	107.1	7.4	135.6

iterations compared to accuracy-based approach, maintaining similar accuracy.

Table II shows a detailed performance analysis of the teacher, student, and training tasks, alongside different OAL solutions. The performance metrics include the GPU's average energy consumption (represented in milliWatt-hour) and the per-frame execution time (measured in milliseconds and Hertz). To assess the efficiency of edge in comparison with server devices, we conducted the performance analysis on two platforms: an NVIDIA Jetson Xavier NX and an Intel i7 desktop equipped with an NVIDIA GeForce RTX 2070 SUPER. Specifically, we evaluated the performance of the teacher model, the student model, the student model with the teacher model running in the background, and the student model with the training process running in the background. This evaluation compares the performance of standalone models with a combination of the student model with the teacher or the training task, that occur in the OAL framework. Parallel processing of student, teacher and training tasks is unfeasible as they exceed the available resources of the Jetson Xavier NX device, then we discarded this option. Additionally, we compared our SVD-based approach with the state-of-the-art approach that differ in the execution rate of the first four configurations.

It is important to note that, in our approach, the repetition of updates matches that of the teacher, as they are executed simultaneously. This is due to the fact that our SVD-based method relies only on the predicted outputs, with the SVD operating at run-time and the teacher executing at training time. Fig. 4 shows the error evolution over time, calculated on marker-based data considered as ground truth.

Fig. 4a shows the accuracy results in the most challenging scenario, characterized by complete dynamism with unrelated actions.

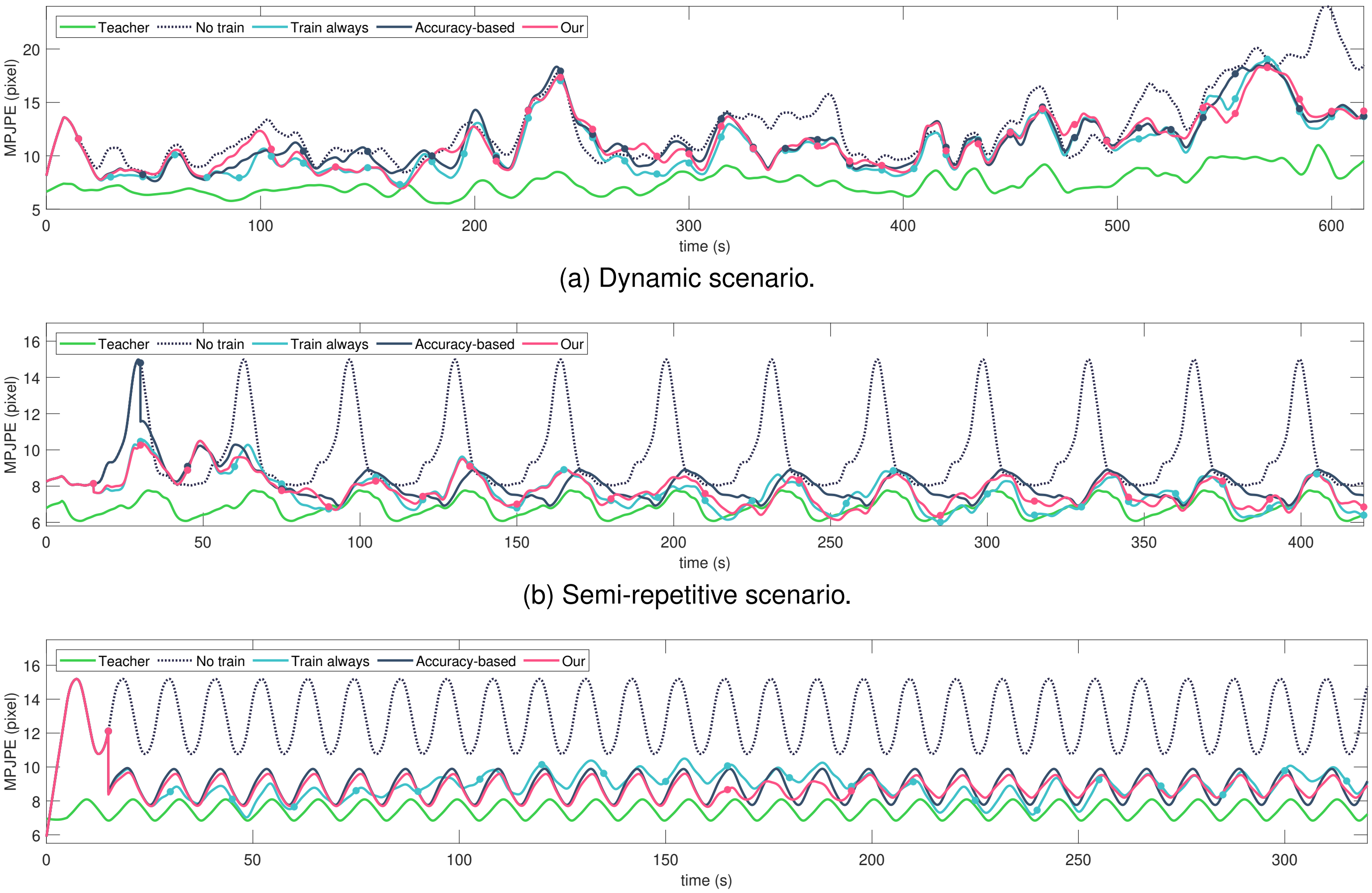
Fig. 4b and Fig. 4c depict respectively the semi-repetitive and the repetitive scenario, which result in similar considerations.

From a performance point of view, our method outperforms the others, avoiding 25.8% of teacher inferences while maintaining an error level comparable to accuracy-based. This result is particularly evident in the last two rows of Table II, where the energy consumption and operating frequency of our OAL approach overcome those of the accuracy-based approach. Our OAL approach outperforms the accuracy-based approach across all scenarios, with the most significant improvement observed in the repetitive scenario, which requires less retraining and, on average, saves a lot of teacher model's execution. Additionally, the operating frequency of our OAL approach is substantially better than that of the teacher model and closely matches the running frequencies of the student model, while maintaining accuracy levels comparable to those of the teacher model.

In the comparison between edge and server performance, the differences in energy consumption and execution time are evident. Our SVD-based training scheduler significantly reduces energy consumption at the edge while maintaining acceptable execution rates. On average, the trade-off between execution frequency and average energy consumption (Hz/mWh) on the NVIDIA Jetson is 1.58, compared to 1.27 on desktop server. This demonstrates the effectiveness of deploying an OAL system entirely at the edge.

The initial model, denoted as "no train", exhibits repeated errors over time. However, the proposed system effectively identifies and mitigates this noise through retraining, resulting in a significant reduction in error. Our training scheduler significantly outperforms the *train always* approach, demonstrating higher efficiency in handling repetitive scenarios. In highly repetitive sequences, continuous training is not only ineffective but also negative, as it reduces the overall accuracy of the network due to overfitting. Our approach succeeds in identifying the optimal number of updates for achieving the best accuracy-updates tradeoff. It conducts less updates compared to "train always" approach but allows more updates compared to accuracy-based methods. While significantly reducing the need for teacher executions, it still obtains acceptable accuracy results.

To assess the robustness of our OAL framework, we also conducted experiments in a real-world setting. It represents



(a) Dynamic scenario.

(b) Semi-repetitive scenario.

(c) Repetitive scenario.

Fig. 4. Effect of different quality assessment approach on the accuracy of the online active learning framework for human pose estimation task.

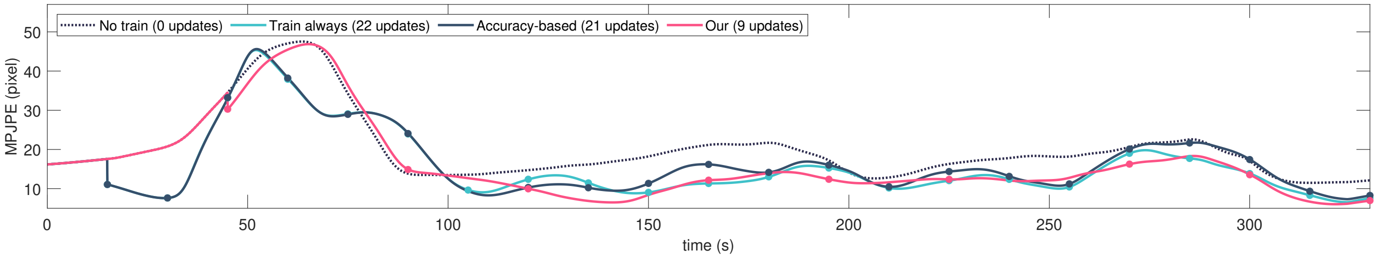


Fig. 5. Real-world case study: the Industry 4.0 scenario.

a typical production line, in which an operator held a semi-static position at his/her workstation, in front of a collaborative robot. The operator behavior corresponds to standard working movements, such as relocating objects from the conveyor belt to carts and viceversa. We captures the videos with a StereoLabs ZED 2 RGB camera (15 FPS, 2K resolution).

Fig. 5 presents the experimental results of the system applied in the real-world scenario. In this case, since there is no true ground truth available, the error is represented as MPJPE from the teacher model. In this challenging scenario, the proposed method demonstrates its effectiveness by minimizing the number of training iterations and significantly reducing the error.

In general, as for the results obtained with the standard dataset, the results obtained with the real-case study confirm the efficiency of the proposed framework. The significant reduction in computational time enhances its applicability on real scenarios. Our system minimizes training iterations while concurrently mitigating errors, particularly in repetitive scenes. This makes the solution well-suited for static scenarios such as workplace environments.

2) Query Strategies: In this paragraph we analyze the query strategy techniques in the case of offline fine-tuning. In addition by varying the types of frame considered, we also varied the number of samples, namely taking the 1%, 5%, 10%, 20% and 40% of the training dataset. Table III reports the accuracy results, also comparing the frame selection techniques with the model without fine-tuning (i.e. 0%) and the model fine-tuned on 100% of the training set.

In the offline fine-tuning evaluation, as we decreased the size of the training dataset, we successfully reduced the computation time from 15 hours to 6 hours, 3 hours, 80 minutes, and 9 minutes namely for the 40%, 20%, 10%, 5%, and 1% of the train dataset.

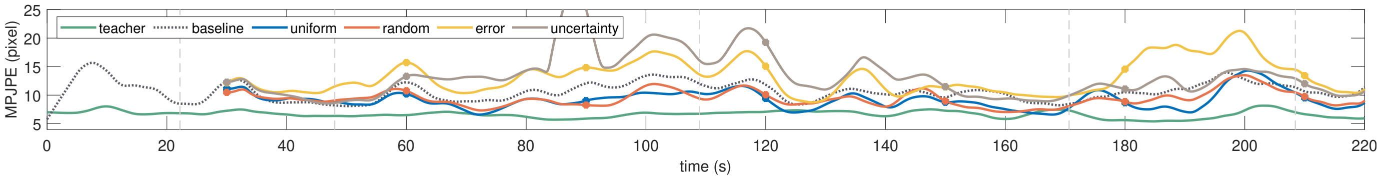


Fig. 6. Comparison between sampling query strategies, all with a sampling rate of 1%.

TABLE III
ACCURACY IN MPJPE (PIXELS) OBTAINED WITH DIFFERENT SAMPLING RATES AND QUERY STRATEGIES.

	0%	**1%**	**5%**	**10%**	**20%**	**40%**	**100%**
uniform	16.33	11.92	**10.49**	10.56	**9.92**	11.52	10.14
random	16.33	11.87	10.73	**10.48**	10.35	**9.81**	10.14
error	16.33	12.48	11.91	10.99	10.58	10.53	10.14
uncertainty	16.33	**11.82**	10.87	10.57	11.13	10.09	10.14

Figure 6 depicts the online fine-tuning evaluation, in which the training process is performed every 30 seconds, according to *train always* apprach. Every training step performs sampling on the last training queue using a sampling percentage of 1% and different sampling metrics. The lines represent the progression of errors during training across different sampling metrics over time. The green line represents Openpose, the black dotted line represents the light model without fine-tuning, and the other lines correspond to the sampling metrics. Both the uniform and random sampling metrics show the best outcomes, whereas the other metrics yield slightly worse results than the baseline.

Active learning can lead to improved results compared to training the model on the entire dataset. The most effective active learning strategies for both offline and continual training are random and uniform, which consistently yield similar outcomes. Generally, these strategies ensure a balanced subset dataset on average, while other approaches may select frames with incorrect labels and false predictions.

B. Results on Object Detection

To test our method in another application domain, where time series of positions are considered, we focused on Object Detection, specifically targeting the detection of cars in time series. This is a widely utilized task at the state-of-the-art. In particular, we employed a YOLOv7 neural network [15] as the teacher and a less accurate SSD (Single Shot MultiBox Detector) [16] as the student. Both networks are designed to perform multi-class, multi-instance detection in a single inference, making them suitable for edge scenarios. The teacher network was pre-trained with the COCO dataset [20], which consists of 80 classes. The student network was trained on the VOC dataset [22], which comprises 20 classes. We aim to specialize our student network for a single class (common to both datasets), namely cars. To achieve this, we utilized sequences containing cars from the LaSOT [23] dataset as the test sequence for our online active learning.

Since both networks are multi-class and multi-instance, we masked the labels related to other classes, thus focusing solely on the multi-instance problem. The terminology ”multi-instance” denotes the presence of multiple vehicles within a single image, leading to the prediction of multiple bounding boxes by both networks. Given that the proposed approach for time series analysis relies on historical data, maintaining temporal consistency for each bounding box is imperative. This entails ensuring that the vehicle identified by box ID 1 at time t aligns with the same vehicle identified by box ID 1 at time t+1. To achieve this, we focused on one car at a time (the most representative, i.e., the car with the highest confidence score from the student network) and applied a tracking algorithm. The tracking algorithm is based on the Jaccard index between bounding boxes belonging to two consecutive frames. We assume that the high working frequency of the camera (i.e., 30 frames per second) is sufficiently high to ensure that two boxes from consecutive frames are in similar positions. The Jaccard index 4, also known as the Jaccard similarity coefficient, is a measure used to quantify the similarity between two sets. It is defined as the size of the intersection of the sets divided by the size of the union of the sets:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4}$$

In our case, A is the bounding box of the car at the instant t and B is the bounding box of the car at the instant $t - 1$ The concept is that the Jaccard index of two bounding boxes corresponding to the same vehicle at two consecutive time instances will be higher compared to the boxes of two different vehicles at two consecutive time instances. Thanks to this, we can associate the bounding boxes corresponding to the same car over time. Each bounding box is represented by two 2D points (in pixels), meaning four coordinates. In this manner, we executed our framework by tuning the parameters through an initial analysis phase. Specifically, r was set to 2 singular values, and the percentage θ to 96. The *sampling* percentage was set to 50%, maintaining uniform sampling. Once our framework determines it is time to conduct training, the teacher model is invoked, which predicts one bounding box for each vehicle present in the image. Subsequently, training proceeds in a manner analogous to that employed for the task of human pose estimation.

In this experiment, we performed an empirical search of training hyper-parameters, in particular the results are performed with 2 epochs, learning rate of 10^{-5} and the Adam optimizer.

The models were evaluated using Mean Absolute Error (MAE), which represents the average absolute error between the vertices of the bounding box of the model under test compared to the ground truth of the dataset. Figure 7 illustrates the time course of model accuracies in a sequence of car scenarios. The teacher model (represented by the green line) shows a lower average error than the student model without fine-tuning (represented by the dashed line). This discrepancy becomes particularly pronounced in the latter part of

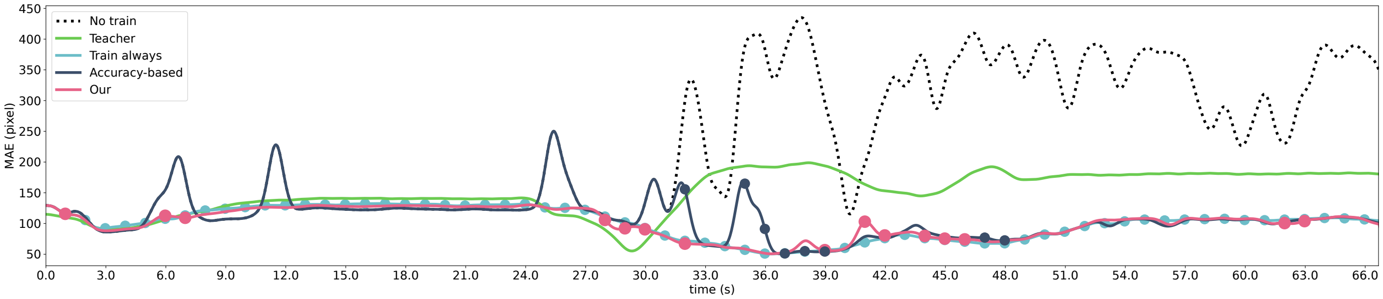


Fig. 7. Effect of different quality assessment approaches on the accuracy of the online active learning framework for the object detection task.

TABLE IV
ACCURACY OF THE OAL PLATFORM VARYING THE TRAINING SCHEDULES IN OBJECT DETECTION TASK.

Metric	**MAE**	**%update**	**%teacher execution**
Teacher	155.5	-	-
No train	245.4	0.0	0.0
Train always	100.2	100.0	100.0
Accuracy-based	108.7	5.3	100.0
Our	100.8	9.2	9.2

TABLE V
ENERGY CONSUMPTION AND LATENCIES OF THE OAL PLATFORM VARYING THE TRAINING SCHEDULES IN OBJECT DETECTION TASK ON AN EDGE DEVICE AND A SERVER.

Device	**Jetson NX**			**RTX 2070 SUPER**		
Performance	**Energy**	**Time**		**Energy**	**Time**	
	mWh	**ms**	**Hz**	**mWh**	**ms**	**Hz**
Accuracy-based	28.2	55.7	17.9	269.3	10.5	95.6
Our	19.3	32.8	30.5	197.4	7.4	135.5

the figure, where the two lines diverge by more than 200 pixels. In the initial segment of the scene, all models show similar performance, as the reference vehicle for the evaluation was stationary at a traffic signal, resulting in a static scene. The *train always* approach (cyan line), unfeasible for online deployment, shows excellent performance, outperforming even the teacher model in the last plot segment. In spite of its counterintuitive nature, whereby student models are trained from the labels of the teacher model, this demonstrates that the model, through its training process, achieves better adaptation to the scene, consequently yielding improved performance. Furthermore, it is noteworthy to mention that in training the SSD model, we employed a data augmentation scheme as proposed by the authors in their paper. The blue line, on the other hand, represents the OAL model inspired by [12]. Although this method is also not achievable in real-time, it performs very well, closely resembling the performance of the *train always* approach, except for some peaks. Our framework performs almost analogously to the *train always* approach, outperforming the accuracy based approach and stands out as the most suitable for deployment in a real-world scenario.

Table IV provides a quantitative measure of the findings depicted in the plot. As mentioned earlier, the teacher model outperforms the student model without training (i.e., *No train*), with a mean absolute error (MAE) of 155.5 pixels compared to 245.4 pixels. Among the methods employing fine-tuning, the *train always* approach shows an improvement, lowering the MAE to 100.2 pixels on average. This methodology, despite its superior performance, demands 100% of the teacher's executions and all feasible training data, making it impractical in a real-world scenario. The accuracy-based approach yields a MAE of 108.7 pixels. It executes the teacher model 100% of the time (for comparison and label acquisition) and conducts training only 5.3% of the time. In contrast, the proposed methodology achieves a better result with a MAE of 100.8 pixels, executing both training and the teacher model (for label acquisition) 9.2% of the time, thereby achieving significant computational savings.

Table V presents a performance analysis similar to that in Table II. This table compares our OAL approach with the accuracy-based approach in terms of energy consumption and execution time on both edge and server devices in OD task. Even for OD task, our OAL approach outperforms the accuracy-based approach, demonstrating higher performance in both execution times and energy. Furthermore, the trade-off between execution frequency and average energy consumption (Hz/mWh) gets better with a deployment in the NVIDIA Jetson (1.58) over a deployment in the desktop server (0.68). These results demonstrate that our approach is generalizable and effective for both human pose estimation and object detection tasks.

V. Conclusion

This work aimed to enable on-device neural network training, with a specific focus on fine-tuning for video analysis models. We developed a novel methodology for determining the optimal moments for initiating training in real-time and privacy-aware systems. Our approach leverages the principles of low rank approximation to identify the presence of noise within a sequence. The execution of the teacher model and subsequent training through knowledge distillation occurs when our approach detects a noisy sequence. While state-of-the-art approaches selected training steps through comparison with teacher predictions, our approach employed low-rank approximation theory on student predictions to determine optimal training scheduling, saving teacher executions. Through comprehensive experimentation, we applied our methodology to two wide-spread tasks (i.e., human pose estimation and object detection) on public datasets and a real case study, demonstrating its effectiveness in achieving high accuracy while saving computational resources. By reducing the frequency of teacher executions, selecting the optimal training samples and minimizing training times, our approach gives a practical solution for online fine-tuning, enabling the deployment of intelligent video analysis applications on resource-constrained edge devices. In future work, we aim to further enhance the efficiency of neural networks, conducting training directly on the device of even more complex models. This includes investigating various accelerators and implementing efficient schedulers to achieve real-time performance.

References

[1] Y. Guo, F. Deligianni, X. Gu, and G.-Z. Yang, "3-D canonical pose estimation and abnormal gait recognition with a single RGB-D camera," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3617–3624, 2019.

[2] J. Lim *et al.*, "Designing path of collision avoidance for mobile manipulator in worker safety monitoring system using reinforcement learning," in *IEEE ICSR*, 2021, pp. 94–97.

[3] M. M. Alam, L. Torgo, and A. Bifet, "A survey on spatio-temporal data analytics systems," *ACM Comput. Surv.*, vol. 54, no. 10s, nov 2022. [Online]. Available: https://doi.org/10.1145/3507904

[4] W. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.

[5] B. Li, B. Wu, J. Su, and G. Wang, "EagleEye: fast sub-net evaluation for efficient neural network pruning," *LNCS*, vol. 12347, pp. 639–654, 2020.

[6] P. Wang, X. He, G. Li, T. Zhao, and J. Cheng, "Sparsity-inducing binarized neural networks," in *Conference on Artificial Intelligence*, 2020, pp. 12 192–12 199.

[7] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, Q. Le, and H. Adam, "Searching for mobilenetv3," in *IEEE ICCV*, 2019, pp. 1314–1324.

[8] S. Teerapittayanon, B. McDanel, and H. Kung, "Branchynet: Fast inference via early exiting from deep neural networks," in *Proc. of International Conference on Pattern Recognition*, vol. 0, 2016, p. 2464 – 2469.

[9] V. Mehlin, S. Schacht, and C. Lanquillon, "Towards energy-efficient deep learning: An overview of energy-efficient approaches along the deep learning lifecycle," 2023. [Online]. Available: https://arxiv.org/abs/2303.01980

[10] T. Panagiotakopoulos, P. L. Dovesi, L. Härenstam-Nielsen, and M. Poggi, "Online domain adaptation for semantic segmentation in ever-changing conditions," in *European Conference on Computer Vision (ECCV)*, 2022.

[11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[12] R. T. Mullapudi, S. Chen, K. Zhang, D. Ramanan, and K. Fatahalian, "Online model distillation for efficient video inference," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-Octob, pp. 3572–3581, 2019.

[13] NVIDIA AI IoT, "Tensor RT Pose Estimation," 2020, https://github.com/NVIDIA-AI-IOT/trt_pose.

[14] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[15] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[17] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 472–487.

[18] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jul 2017, pp. 2261–2269. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.243

[19] J. Redmon, "Darknet: Open source neural networks in c," http://pjreddie.com/darknet/, 2013–2016.

[20] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.

[21] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[23] H. Fan, H. Bai, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, Harshit, M. Huang, J. Liu, Y. Xu, C. Liao, L. Yuan, and H. Ling, "Lasot: A high-quality large-scale single object tracking benchmark," 2020. [Online]. Available: https://arxiv.org/abs/2009.03465

[24] J. Xu, W. Zhou, Z. Fu, H. Zhou, and L. Li, "A survey on green deep learning," 2021. [Online]. Available: https://arxiv.org/abs/2111.05193

[25] P.-Y. Lu, C.-L. Li, and H.-T. Lin, "Re-benchmarking pool-based active learning for binary classification," 2023.

[26] B. Liu and V. Ferrari, "Active learning for human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4363–4372.

[27] D. Yoo and I. S. Kweon, "Learning loss for active learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 93–102.

[28] W. Zhang, Z. Guo, R. Zhi, and B. Wang, "Deep active learning for human pose estimation via consistency weighted core-set approach," in *2021 IEEE International Conference on Image Processing*, 2021, pp. 909–913.

[29] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," *Advances in Neural Information Processing Systems*, 2020.

[30] M. Shukla, R. Roy, P. Singh, S. Ahmed, and A. Alahi, "Vl4pose: Active learning through out-of-distribution detection for pose estimation," in *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. [Online]. Available: https://bmvc2022.mpi-inf.mpg.de/0610.pdf

[31] D. Cacciarelli and M. Kulahci, "A survey on online active learning," *arXiv preprint arXiv:2302.08893*, 2023.

[32] D. Manjah, D. Cacciarelli, B. Standaert, M. Benkedadra, G. R. de Hertaing, B. Macq, S. Galland, and C. De Vleeschouwer, "Stream-based active distillation for scalable model deployment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4998–5006.

[33] M. Khani, P. Hamadanian, A. Nasr-Esfahany, and M. Alizadeh, "Real-Time Video Inference on Edge Devices via Adaptive Model Streaming," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 4552–4562, 2021.

[34] S. Cammarasana and G. Patane, "Spatio-temporal analysis and comparison of 3d videos," *The Visual Computer*, vol. 39, 04 2022.

[35] Cambridge University Press, Jan. 2019, p. 3–46. [Online]. Available: http://dx.doi.org/10.1017/9781108380690.002

[36] N. B. Erichson, S. L. Brunton, and J. N. Kutz, "Compressed singular value decomposition for image and video processing," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. IEEE, Oct. 2017. [Online]. Available: http://dx.doi.org/10.1109/ICCVW.2017.222

[37] J. Chai and J. K. Hodgins, "Performance animation from low-dimensional control signals," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 686–696, Jul. 2005. [Online]. Available: https://doi.org/10.1145/1073204.1073248

[38] K. Miura, H. Furukawa, and M. Shoji, "Similarity of human motion: congruity between perception and data," Oct. 2006. [Online]. Available: https://doi.org/10.1109/icsmc.2006.384875

[39] M. Ding and G. Fan, "Multilayer joint gait-pose manifolds for human gait motion modeling," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2413–2424, 2014.

[40] R. Lai, P. Yuen, and K. Lee, "Motion capture data completion and denoising by singular value thresholding," *Proc. Eurographics Assoc.*, pp. 1–4, 2011. [Online]. Available: http://www.comp.hkbu.edu.hk/{~}yqlai/images/egfinal.pdf

[41] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Transactions on Automatic Control*, vol. 25, no. 2, pp. 164–176, 1980.

[42] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936. [Online]. Available: https://doi.org/10.1007/bf02288367