Pose-guided token selection for the recognition of activities of daily living

Ricardo Pizarro[a], Roberto Valle[b], José M. Buenaposada[c], Luis M. Bergasa[a], Luis Baumela[b]

[a]*Departamento de Electrónica, Universidad de Alcalá, Alcalá de Henares, Spain*
[b]*Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Madrid, Spain*
[c]*Departamento de Informática y Estadística, Universidad Rey Juan Carlos, Móstoles, Spain*

Abstract

Large pre-trained video transformers are becoming the standard architecture for video processing due to their exceptional accuracy. However, their quadratic computational complexity has been a major obstacle to their practical application in problems that require the recognition of precise motion patterns in video, such as in the recognition of Activities of Daily Living (ADL). Techniques like token pruning help mitigate their computational cost, but overlook some specific aspects of this task such as the actor movement. To address this we propose an improved token selection method that integrates semantic information from the ADL recognition task with that of human motion. Our model relies on a multi-task architecture that infers human pose and activity classification from RGB images. We show that guiding token pruning with motion information significantly improves the trade-off between higher efficiency, obtained by reducing the number of tokens, and accuracy of the classification task. We evaluate our model on three popular ADL recognition benchmarks with their respective cross-subject and

cross-view setups. In our experiments, a video transformer modified with our proposed modules sets a new state-of-the-art on the ADL recognition task whilst achieving significant reductions in computational cost.
Keywords: activities of daily living recognition, efficiency in transformers, token selection, motion heatmaps

1. Introduction

Activities of Daily Living (ADL) encompass the fundamental tasks of daily life, such as eating, cooking, and managing medications. They play a crucial role in assessing a person's ability to function independently. Their recognition is used to monitor the elderly or people with disabilities and to evaluate their functional ability in conditions such as dementia, stroke, or aging. The models and techniques of computer vision used to recognize them share similarities with the broader field of human action recognition. However, ADLs present specific challenges, such as the existence of short and subtle actions that exhibit a similar visual appearance but differ in motion [1]. This requires the precise analysis of human body motion patterns within videos' spatio-temporal context.

In the recognition of human actions we have seen a transition from methods using CNNs [2, 3, 4] and 3D-CNNs [5, 6, 7] or a mixture of both [4] to transformers [8, 9, 10]. Using self-supervised learning techniques and the use of large-scale datasets, recent video transformer models achieve the highest accuracy on the human action recognition problem [11]. A key limitation in using these models to analyze video is their quadratic complexity, which increases the computational demands as the number of spatio-temporal tokens

grows. Although progress has been made in this area, there is still considerable room for improvement, especially for recognizing subtle motions and when the trade-off between accuracy and efficiency is of practical relevance. Both are crucial ingredients in making the recognition of ADL a household product. Applications such as falling detection or ensuring that medication is taken correctly demand real-time performance, making computationally expensive models impractical.

One technique to achieve a better trade-off between accuracy and efficiency is token selection, where a percentage of tokens are discarded at certain blocks within the transformer model, reducing the total number of tokens in the model. Popular techniques include Top-K [12], where token selection is guided by keeping the K tokens with the greatest attention to the class token, merging similar tokens [13], or a mixture of both [14, 15, 16]. However, these techniques often lack consideration for factors such as human pose and its temporal dynamics. This can lead to suboptimal performance in ADL scenarios that require a nuanced understanding of human actions, resulting in a potential loss of critical information.

In this paper, we present a token selection method for transformer models that integrates semantic information from both the activity recognition task and human motion. We aim to improve the attention of the transformer on the actor's motion and, at the same time, reduce computational requirements of the model. Our module can be integrated on ViT-based architectures such as InternVideo2 [18] and VideoMAEv2 [11]. These transformer architectures are pre-trained with a self-supervised strategy and refined with a large human action database. Our method, called PO-GUISE, is trained in a multi-task

(a) Baseline ViT model

(b) PO-GUISE ($\rho = 0.1, \lambda = 0.1$)

Figure 1: Attention maps for the "Drink.Frombottle" action on Toyota-Smarthome (CS) [17]. Colored rectangles represent the attention weight assigned by each visual token to the classification token, lighter yellow rectangles indicating a low attention from that token. PO-GUISE concentrates attention on task-relevant regions, improving computational efficiency by discarding irrelevant tokens.

fashion using RGB videos. They are converted into spatiotemporal visual tokens and are processed alongside heatmap tokens representing temporal representations of human poses. We have extended the traditional heatmap to predict the motion of the keypoints of multiple actors in video. Our token selection method prunes spatiotemporal visual tokens, referred to as *visual tokens*, that do not pay enough attention to semantic tokens, those relevant to human motion and action recognition. To ensure that information is not lost during pruning, our merging method summarizes the pruned tokens by averaging similar dropped tokens. Fig. 1 shows that our method selects tokens primarily on the actor, while the baseline model focuses on potentially irrelevant parts of the scene. To our knowledge, we are the first to improve the accuracy of transformer models for ADL recognition while reducing its computational cost using human pose and motion information. Moreover, our approach does not require an external keypoint detection model. In summary, we pioneer the introduction of human motion information into the token selection process in the video transformer architecture.

The contributions of our work are as follows.

- A token selection method guided by human motion and class information tailored to the recognition of activities of daily living. Focuses the attention of the model on the motion of the actor and improves the trade-off between efficiency and accuracy compared to other methods from the state-of-the-art, even at very low token keep rates.

- A representation of human motion based on a feature map shared by all body keypoint temporal heatmaps, that is agnostic of the number of people in the scene and allows our method to be used on multi-actor

datasets.

- Our method sets a new state-of-the-art in various activities of daily living RGB video benchmarks, while being much more efficient than other top performing methods based on video transformers.

2. Related work

In this section, we review the human action recognition and activities of daily living literature. Recognizing actions in videos requires considering variations in the location and poses of actors within the scene, as well as their movement.

Human Action Recognition and ADL. One way to analyze motion in videos is to compute convolutions in both the image and the time dimensions with 3D CNNs [5]. A popular approach is the two-stream CNN [2, 3, 4] that uses both RGB and optical flow maps. However, optical flow only gives short temporal scale information. More recent work use a Recurrent Neural Network (RNN) [19] on top of a two-stream network [3] to process a longer but still limited temporal context. The adoption of video transformers in action recognition allows for a holistic temporal context to be established [8, 9, 10], although with quadratic complexity in the number of visual tokens.

The human pose and its realization in the form of probability maps, or heatmaps, corresponding to the location of body keypoints has proven to be very discriminative in action recognition [20, 21, 22, 23, 24, 25, 26, 27]. Many previous studies have used an external human pose estimation model [22, 21, 23, 28, 29, 30, 31]. This is also the case with recent transformer based methods [25, 26, 27]. Having an external pose estimation model not only

increases the computational cost but also decreases the system robustness in situations where the external model fails. Few methods adopt a multi-task strategy to estimate pose and recognize actions in the same model [19, 32]. A recent approach achieves top performance in the recognition of activities of daily living by combining 2D and 3D human pose [10]. In our solution we also adopt a multi-task strategy. However, unlike these approaches, we use human pose to select the most informative video tokens by guiding the model's attention to human motion, while reducing the computational requirements of the model.

Computational requirements of Video Transformers. The quadratic complexity in the number of tokens in a transformer is a fundamental limitation for its use in real-time video analysis. This problem can be addressed in different ways. Some methods modify the attention mechanism itself to reduce this quadratic complexity. For example, one approach is to factorize attention along the spatial and temporal dimensions [33]. Another is to restrict attention to small local windows and shift these windows hierarchically [34].

Another approach is token selection, in which a dedicated mechanism prunes or merges the visual tokens processed by the network, discarding those considered irrelevant to the task. This is achieved while preserving the integrity of the transformer's weights and underlying architecture.

Token selection methods can be categorized into pruning or merging strategies. Token pruning methods focus on identifying and removing less informative tokens. EViT [14], which uses a Top-K approach, selects the K tokens with the highest attention to the class token, where the non-selected

tokens are fused into one token. PPT [35], introduces a learnable token per body keypoint and uses their attention values to prune visual tokens. The main limitation of PPT is the fixed number of keypoint tokens used in training, which limits the number of actors in the scene. EVAD [9], leverages attention to visual tokens on a key-frame to determine which tokens to retain. The TPS (Token Pruning and Squeezing) module [15], is a module for image transformers. It uses a Top-K token pruning step and a squeeze step that merges the non-selected tokens into the selected ones via matching and similarity-based fusing. Another form of guiding pruning from image information is based on patches, where inter-patch attention and dynamic pruning are applied to take advantage of the rich structure of the patch relations [36].

Token merging techniques combine similar tokens to reduce redundancy, such as ToMe [13], which merges similar tokens, as dictated by their cosine similarity, into new ones. DTMFormer [37], which adaptively clusters tokens into fewer semantic tokens via an attention-guided mechanism. Another technique is a partitioned token fusion and pruning strategy. It discards low-correlation background token information and fuses medium-correlation token. This technique has been applied to the field of object tracking [16].

Haurum et al. [12] provides a systematic comparison of ten popular token reduction methods, finding that pruning-based methods such as Top-K and EViT [14] consistently perform best.

However, a significant limitation of existing token selection methods is their lack of task-specific considerations. Specifically for the ADL task, these methods do not account for the human pose and its temporal dynamics directly, potentially resulting in the loss of crucial information.

Our proposal. We present a novel token selection method guided by both temporal human pose heatmaps and ADL. We use a multi-task strategy, estimating both human motion heatmaps and activity, which differs from the usual and less efficient approach using externally provided landmarks [25, 26, 27]. Also, and differently to π-Vit [10], our approach uses the estimated motion to reduce the number of spatio-temporal tokens. This strategy focuses the attention of the model on the actor's movements and reduces the computational complexity of the transformer. As a result, it maintains or even enhances the accuracy of the baseline model. In addition, its accuracy decreases much more slowly than that of other token selection methods at very low computational budgets. Compared with the baseline model, PO-GUISE in default settings reduces computation by a remarkable 30% and improves the accuracy by 0.55%, 1.74% and 3.84% in the NTU60, NTU120 and Toyota-Smarthome datasets, respectively, in the cross-subject protocol (see Tables 5 and 4).

3. POse-GUIded multi-task video transformer with token SElection (PO-GUISE)

Our approach incorporates a pre-trained video transformer [11, 18] as its encoding mechanism. The video transformer is fine-tuned in different action recognition datasets. To facilitate human body keypoints localization and guide our token selection, we have integrated the pose heatmaps prediction and action classification tasks. Additionally, to mitigate the computational demands associated with video transformer models, we introduce the PO-GUISE module, which effectively reduces the number of visual tokens.

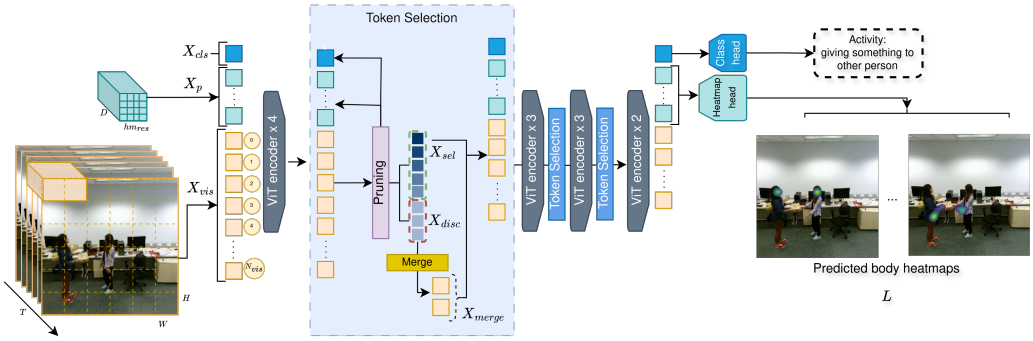


Figure 2: Our architecture consists of 4 stages. An input clip is tokenized and processed by a ViT encoder alongside learnable class and heatmap tokens. Our token selection module is inserted in the first three stages of the ViT encoder, reducing the number of tokens after each stage. The model outputs both the activity classification and the corresponding motion heatmaps.

A comprehensive visual representation of our model is given in Fig. 2. In the following sections, we provide a detailed explanation of each component within our model.

3.1. Video Transformer and human-pose processing

Consider a video segment, or clip, with dimensions $T \times C \times H \times W$ where T is the number of frames and C, H, W are the channels, height, and width of each frame, respectively. In our experiments, we define $T = 16$, $C = 3$, $H = 224$ and $W = 224$ respectively. To process a clip with a video transformer [11], we use the joint space-time cube embedding [33]. This technique samples non-overlapping cubes from the input video clip, which are then fed into the embedding layer. It divides the input video tensor into cubes of dimension $2 \times C \times 16 \times 16$, resulting in a set of $N_{vis} = t \cdot h \cdot w$ visual tokens, where $t = \frac{T}{2}, h = \frac{H}{16}, w = \frac{W}{16}$. We then project tokens to D

dimensions using a linear embedding layer, resulting in an input tensor with shape $X_{vis} \in \mathbb{R}^{N_{vis} \times D}$. Next, we apply a positional embedding to each token, and a learnable class token, $X_{cls} \in \mathbb{R}^{1 \times D}$, is concatenated to the sequence. For the computation of human-pose heatmaps, our model incorporates $N_p = hm_{res} \cdot hm_{res}$ learnable tokens into the input sequence defined as $X_p \in \mathbb{R}^{N_p \times D}$, where hm_{res} defines the heatmap feature map resolution and total number of tokens it is represented by. The complete sequence of tokens, including the class, pose and visual tokens $X = (X_{cls}, X_p, X_{vis}) \in \mathbb{R}^{N \times D}$ where $N = 1 + N_p + N_{vis}$, is then processed using a standard ViT architecture. The transformed class token X_{cls} is used in a multilayer perceptron (MLP) for the classification task, while the X_p pose tokens are passed through a heatmaps estimation head to be compared against the ground truth heatmaps for pose estimation (one heatmap per human body keypoint).

3.2. Human-pose estimation task

A crucial part of our approach involves the use of temporal heatmaps, which enhance the training process and facilitate token selection. These heatmaps are derived from learnable tokens, similar to those in PPT [35]. However, our method further refines PPT's image-only processing by extending its capabilities to handle a variable number of keypoints, video inputs, and multi-person heatmap predictions.

Heatmap prediction starts with the introduction of additional tokens to the network, X_p. After passing through the encoder, these tokens are processed by a lightweight decoder (Heatmap head) to convert the tokens into heatmaps. The architecture of the Heatmap head consists of two deconvolution layers followed by a convolution layer with a 1×1 kernel and with

Figure 3: Motion heatmap generation. We aggregate the movement of the keypoints through time into a single heatmap. The figure shows, from left to right, the same keypoint at three different points in time and the corresponding aggregated heatmap.

output channels equal to the number of landmarks L [38]. The output of this decoder is then directly compared with the ground truth heatmaps by measuring the mean-squared error.

While these tokens are inherently capable of predicting heatmaps for an individual frame within a video clip, we can adapt them to capture the entire sequence of movements by modifying the ground truth labels. The use of heatmaps instead of coordinate representations provides greater flexibility by allowing the incorporation of additional information directly within the heatmaps, without requiring any structural changes to the network architecture. We generate time-aware heatmaps by averaging the spatial heatmaps from the ground-truth labels, a Gaussian centered at the location of each annotated landmark, across the whole video clip. It results in a ground truth heatmap where each keypoint movement within the clip is visible. Likewise, the framework can be extended to predict multi-person heatmaps by combining detection data from multiple individuals inside a single heatmap. In Fig. 3 we show an example motion heatmap for the multi-actor case.

3.3. POse-GUIded token SElection module

The use of joint space-time cube embeddings for processing videos is computationally expensive, which is not ideal for use in environments with limited computing power. Videos naturally contain repetitive information over time and areas with no information for action recognition. Thus, we propose the use of token pruning to reduce computation without losing important content.

We introduce a novel approach named PO-GUISE. This method leverages the informative content of the class and heatmap tokens to improve the token selection process. Furthermore, to prevent the loss of potentially valuable information, PO-GUISE also merges some of the tokens that were not initially selected during the pruning step. This merging step is crucial as it compensates for any potentially relevant data that might not have been identified by the pruning algorithm. Fig. 2 shows an overview of this two-step token selection.

We integrate our token selection module into the transformer network architecture at specific intervals. The ViT base architecture consists of 12 layers, we divide these in 4 stages, where each stage consists of 4,3,3,2 layers, respectively. We place the module at the output of each of the first three stages. This results in a total of three token selection layers within a ViT-base model (see Fig. 2). In doing so, our goal is to strike a balance between reducing computational load and maintaining the critical information necessary to efficiently process the video.

3.3.1. Token pruning.

Building upon existing token pruning methods like EVIT [14] and EVAD [9], our approach introduces a novel integration of spatial information. Specifically, we leverage heatmap tokens to guide attention towards visual tokens that correspond to actor locations. Let $\mathcal{A}_M \in \mathbb{R}^{M \times N_{vis} \times (1+N_p)}$ be the attention tensor from M heads, obtained from processing the tokens in $X \in \mathbb{R}^{N \times D}$, and then indexing by the attention the visual tokens (X_{vis}) give to the heatmap (X_p) and class (X_{cls}) tokens. We average across attention heads to condense it into an $N_{vis} \times (1 + N_p)$ matrix, resulting in $\mathcal{A}_{vis} \in \mathbb{R}^{N_{vis} \times (1+N_p)}$, see Fig. 4. We then multiply by a small constant factor κ, the class attention scores and by $1-\kappa$, the heatmap token attention scores to denote the relative importance between them. Next, by summing the rows of $\mathcal{A}_{vis}$, we get a vector of token scores, $\mathcal{T} \in \mathbb{R}^{N_{vis}}$. Each element in this tensor reflects the aggregated importance of a visual token influenced by the attention to the semantic tokens, (X_{cls}, X_p). The final pruning decision is based on these aggregated scores, allowing us to retain visual tokens that are deemed most significant in the context of both global class information and local spatial heatmap cues. The computed attention score for the i-th visual token can also be formulated as:

$$\mathcal{T}(i) = \mathcal{A}_{vis}(i, 0) \cdot \kappa + \left(\sum_{j=1}^{N_p} \mathcal{A}_{vis}(i, j) \right) \cdot (1 - \kappa),$$

where $\mathcal{A}_{vis}(i, j)$ is the attention score from i-th visual token to j-th semantic token, and κ is a constant factor to balance the importance between class and heatmap tokens.

We use $\mathcal{T}$ to select the N_{sel} most significant tokens, based on their calcu-

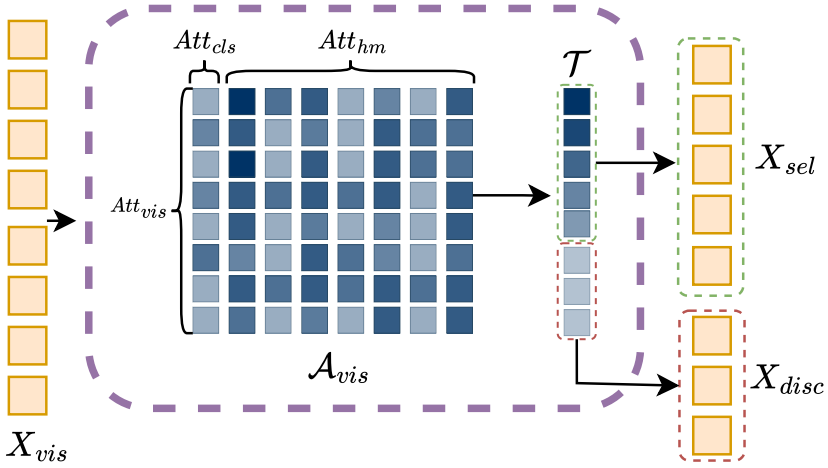


Figure 4: Token pruning diagram. The attention obtained from X_{vis} guides the token pruning. Each row in $\mathcal{A}_{vis}$ corresponds to the attention a visual token (Att_{vis}) gives to the class (Att_{cls}) and heatmap (Att_{hm}) tokens. The Top-K tokens with most attention ($\mathcal{T}$) are selected as output of the step, while the non-selected go through a merging step.

lated scores. The number of selected tokens is determined by $N_{sel} = N_{vis} \cdot \rho$, where the keep rate ρ is a predefined threshold in the range (0, 1]. Resulting in a set of selected tokens, $X_{sel} \in \mathbb{R}^{N_{sel} \times D}$, and a set of discarded ones, $X_{disc} \in \mathbb{R}^{(N_{vis} - N_{sel}) \times D}$. X_{sel} which will be processed in the next network block. Fig. 4 illustrates an overview of the pruning step.

3.3.2. Token merging.

The process of token pruning might exclude information that is important for later processing stages, or information that is not immediately apparent from examining the attention between classes and the associated heatmaps. To mitigate this, we introduce a token merging phase for the discarded tokens,

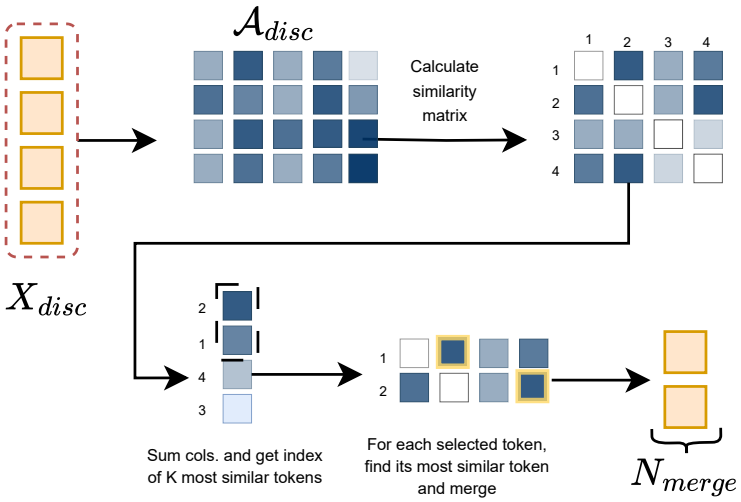


Figure 5: Token merging diagram. The discarded tokens from the previous pruning step are merged by their similarity. The similarity between tokens is measured by their attention to each other (A_{disc}). The N_{merge} most similar tokens are selected and then merged with their corresponding most similar token.

X_{disc}. This phase employs cosine similarity to identify and merge tokens with highly aligned features. Our approach adapts the merging strategy of ToMe [13] by implementing an alternative matching algorithm that is better suited to our context. Unlike ToMe, which initially partitions tokens into two sets, our algorithm is more flexible, allowing the merging of an arbitrary number of tokens. The number of output tokens in this phase is controlled by $N_{merge} = N_{disc} \cdot \lambda$ with λ being a predefined threshold in the range (0, 1]. Fig. 5 shows an overview of the merging method.

This phase begins with the use of the attention tensor A_{disc} obtained from X_{disc}. We then use A_{disc} to compute the pairwise cosine similarity for these tokens, generating a similarity matrix $S \in \mathbb{R}^{N_{disc} \times N_{disc}}$. The diagonal elements of S are masked to prevent the tokens from merging with themselves. Each row of S represents the similarity of a specific token to all other tokens within A_{disc}.

Next, for each token in X_{disc}, we identify its merge candidate as the token with the highest cosine similarity, according to the respective row in S. Subsequently, we select the N_{merge} tokens that exhibit the strongest similarity to their respective candidates. This selective aggregation ensures that the information from tokens with substantial similarity is preserved. These selected tokens are then merged with their corresponding candidates by averaging their feature vectors, resulting in a new set of tokens, $X_{merge} \in \mathbb{R}^{N_{merge} \times D}$. Finally, X_{merge} and X_{sel} are concatenated to be processed by the next network block. This process ensures that potentially relevant information is not lost and is passed on to subsequent layers. A detailed description of this module can be found in Algorithm 1.

4. Experiments

In this section, we evaluate our multi-task video transformer. In all experiments *HM(P)* stands for spatio-temporal heatmaps computed for multiple-person poses *PR* stands for the use of token pruning by: C using attention to the class token; MF using attention to the middle frame visual tokens; or P using attention to the tokens used to compute human motion heatmaps. MG stands for our proposal to merge pruned tokens. PO-GUISE corresponds to

Algorithm 1 Token Merging

$X \in \mathbb{R}^{N \times D}$: Original feature tensor
$F \in \mathbb{R}^{N' \times D}$: Feature tensor of unselected tokens
k: Number of tokens to merge based on similarity
$F_{merged} \in \mathbb{R}^{M \times D}$: Merged feature tensor
$S \in \mathbb{R}^{N' \times N'}$: Similarity matrix
// Compute cosine similarity for discarded tokens
for $i = 1$ to N' **do**
    for $j = 1$ to N' **do**
        $S_{ij} \leftarrow \frac{F_i \cdot F_j}{\|F_i\| \|F_j\|}$ ▷ Cosine similarity
    end for
end for
$S \leftarrow S - \text{diag}(\text{diag}(S))$ ▷ Set diagonal to zero
// Identify merge candidates based on similarity
for $i = 1$ to N **do**
    $merge\_candidate[i] \leftarrow \textsc{Max}(S_{i,:})$
end for
// Select the top-k most similar tokens based on S
$merge\_candidate \leftarrow \text{sort}(merge\_candidate)[:k]$
// Merge source tokens with the selected ones by
$F_{merged} \leftarrow \text{mean}(X[merge\_candidate], axis = 0)$
return F_{merged}

adding +HM(P)+PR(C+P)+MG to the baseline video transformer. Within each experiment, the results of the model in the first, second and third positions are shown, respectively, in bold, underline or double underline.

4.1. Datasets

We use popular ADL recognition datasets for evaluation: NTU60 [5], NTU120 [39], and Toyota-Smarthome [17]. We employ two standard evaluation protocols established in the datasets, cross-subject (CS) and cross-view (CV) or cross-set (CSet). In the CS protocol, the training and testing sets are split according to the identity of the subject, ensuring that there is no overlap between actors. In the CV or CSet protocol, different camera viewpoints are used for training and testing, while all subjects are included in both sets. We present the overall accuracy (*Acc.*) or the average-per-class accuracy (mean class accuracy, mCA) when appropriate due to the class imbalance present in some datasets.

NTU120 is a large-scale human action recognition data set for activities of daily living. It features 114K videos, multiple camera views, 106 subjects, and 120 different classes. We follow the cross-subject protocol (CS), where train-test sets feature different subjects, and cross-setup (CSet) protocol which uses different camera setups in training and testing. The NTU60 dataset is a subset that contains only 57K videos, 40 subjects, and 60 classes. We follow the CS and CV protocols. For both NTU datasets we report the overall accuracy (*Acc.*).

Toyota-Smarthome is a dataset for activities of daily living performed by seniors. The dataset consists of 16K RGB clips of 31 activity classes performed by 18 subjects and 7 different camera viewpoints. We evaluate using

the cross-subject (CS) protocol with 31 classes. We also use two cross-view protocols, CV1 and CV2, both of which use a 19-class subset and cameras 2 and 5 for testing and validation, respectively. For training, CV1 uses only camera 1 while CV2 uses cameras 1, 3, 4, 6, and 7. We report the mean class accuracy (mCA).

4.2. Implementation details

Unless otherwise stated, we use a ViT-base model with pre-trained weights from VideoMAEv2 [11]. These have been distilled from the pre-trained ViT-giant model *vit_b_k710_dl_from_giant*. For classification, we use cross-entropy loss and log-scaled MSE for heatmap prediction. We also use Nash-MTL [40] to balance both tasks. We set the heatmap resolution to $hm_{res} = 8$. We use the AdamW [41] optimizer with a Cosine Annealing learning rate scheduler [42]. Data augmentation includes Cutmix [43] (CMx), Mixup [44] (MxU) and RandAug [45]. For our PO-GUISE model, we set pruning keep rate to $\rho = 0.6$ and merge keep rate to $\lambda = 0.3$ in all experiments unless otherwise stated.

All of our experiments are done on an NVIDIA DGX server with 4 A100-80GB GPUs. Training is done using Pytorch 2.3 [46], and a hyperparameter search is done on the learning rates using Wandb [47] with a Bayesian search on validation loss.

For both NTU120 and NTU60 we follow the official implementation, discarding the examples where no pose was recorded. The detailed hyperparameters used for the experiments in NTU60, NTU120, and Toyota-Smarthome can be seen in Table 1.

At inference we crop the central part of the frame in NTU with full height,

Configuration	Toyota-SM (CV)	NTU/Toyota-SM All/(CS)
Pre-trained weights	*vit_b_k710_dl_from_giant*	
MSE scaling factor	1000	
Learning rate backbone	0.00007	0.0001
Learning rate heads	0.0003	0.0006
Optimizer	Adamw	
Learning rate scheduler	Cosine Annealing	
RandAug. M	7	
RandAug. N	4	
label smoothing	0.1	
CMx & MxU prob.	1.0	
CMx & MxU switch prob.	0.5	
Gradient clipping	1.5	
accumulate_grad_batches	2	
Batch size	16	
Merge feat. sim. matrix	Attention	
Epochs	350	
Early Stopping	30	
#Landmarks	13	25/13
PO-GUISE ρ	0.6	
PO-GUISE λ	0.3	

Table 1: Training parameters used in the main paper experiments.

keeping the aspect ratio and resizing it to 224×224 pixels and each labeled clip was sampled uniformly over time. With Toyota-Smarthome we use the same cropping strategy as in NTU. We follow the official implementation and temporally divide each labeled clip into 4-second samples (128 frames). We reach the final classification by averaging the logits of the samples from each clip.

4.3. Ablation study

For the ablation experiments (see Table 2), we use the Toyota-Smarthome and NTU60 datasets following in both cases their cross-subject procedure. Our baseline result is obtained by fine-tuning a state-of-the-art video transformer, VideoMAEv2 [11] pre-trained in Kinetics [4]. The accuracy for the baseline is 73.14% and 94.29% in Toyota-Smarthome and NTU60, respectively.

4.3.1. Comparison with baseline.

First, we test the baseline plus semantic information in the form of a human pose estimation task, see baseline+HM(P) in Table 2. On average, it increases the accuracy of all actions by 2.87 and 0.18 points in Toyota-Smarthome and NTU60, respectively. Pose information provides a significant improvement in the accuracy of some actions. A small drawback is the increased computational cost of 5% more GFLOPS, due to the extra tokens that need to be processed for the human pose estimation.

We also compare different methods of token selection from the state-of-the-art on the baseline model while maintaining similar GFLOPS for each experiment. We test Top-K pruning by attention to the class token [12],

Method	Toyota-SM mCA. (↑)	NTU60 Acc. (↑)	GFlops (↓)
VideoMAEv2-base (baseline)	73.14	94.29	360
+PR(C)	73.30	93.45	232
+PR(MF)	70.77	94.09	232
+PR(C)+MG	73.89	94.10	232
+HM(P)	76.01	94.47	379
+HM(P)+PR(C)	74.94	93.93	249
+HM(P)+PR(C+P)	75.41	94.57	249
+HM(P)+ToMe	73.80	88.35	190
+HM(P)+PR(C+P)+ToMe	74.65	93.84	249
+HM(P)+PR(C+P)+MG	**76.98**	**94.84**	249

Table 2: Ablation study. Test results on Toyota-Smarthome (CS) and NTU60 (CS) using different model configurations. VideoMAEv2-base is the baseline experiment and the rest are independent experiments adding something to baseline.

baseline+PR(C), pruning by attention to the middle frame visual tokens [9], baseline+PR(MF), and adding our token merging solution to the class token pruning, baseline+PR(C)+MG. We find that for all configurations there is a loss in accuracy when compared to the baseline. In Toyota-Smarthome, utilizing PR(MF), similar to the method in EVAD [9], resulted in a larger loss in accuracy than with PR(C), -2.37% vs +0.16%. This means that the visual tokens in the middle frame are not as informative compared to relying only on the class token for token selection. The use of PR(C)+MG resulted in a small performance gain of 0.75% in Toyota-Smarthome while in NTU60 we obtain a small reduction of 0.19%. This suggests that merging tokens is beneficial in preserving valuable information that pruning alone may not capture. This is crucial for maintaining model accuracy while increasing computational efficiency. Note here that token pruning reduces GFLOPs by 35% (360 to 232) and merging does not add a significant amount of processing.

The last set of experiments in Table 2 assesses the influence of different token selection methods in the multi-task model, baseline+HM(P). The first interesting result is that pruning guided by the class token, baseline+HM(P)+PR(C), affects the performance of the model, 1.07% and 0.54% less accuracy than baseline+HM(P) for both Toyota-Smarthome and NTU60. However, we found that our token pruning guided by class and pose tokens, baseline+HM(P)+PR(C+P), outperforms pruning based solely on class information, baseline+HM(P)+PR(C), by 0.47% and 0.64%. In addition, employing the entire PO-GUISE model (baseline+HM(P)+PR(C+P)+MG) yields an additional improvement of 2.04% and 0.91% over PR(C). We perform additional experiments to compare with the ToMe merging method [13].

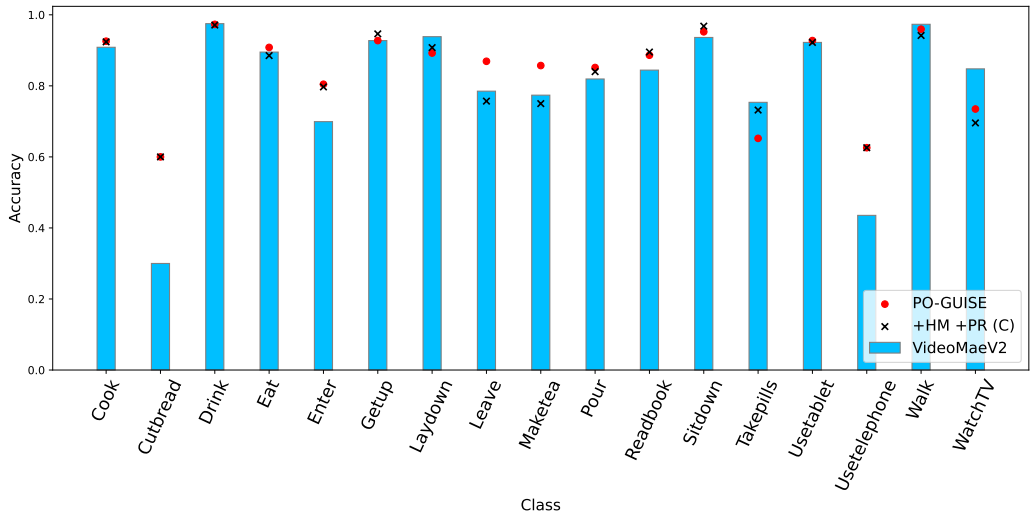


Figure 6: Per-class accuracy comparison on Toyota-Smarthome (CS). We show results for the baseline model (VideoMAEv2-base), Top-K pruning (PR(C)), and PO-GUISE. We have merged some classes for an easier visualization.

The combination of baseline+HM(P)+PR(C+P)+ToMe shows a reduction of 2.33% in accuracy compared to PO-GUISE with our token merging procedure. Lastly, PO-GUISE model achieves a reduction in GFLOPS around 34% while also increasing the accuracy by 0.97% and 0.37% over the baseline+HM(P). These results highlight the effectiveness of pose-guided pruning and the merging process in efficiently selecting task-relevant tokens. In Fig. 6 we show the per-class-accuracy of our method against the baseline model and the Top-K (PR(C)) pruning technique. PO-GUISE obtains an improvement across virtually all classes. The improvement is most notable in classes that require the recognition of fine-grained actions, such as "Use telephone," "Cut bread," and "Make tea," where our method significantly outperforms the baseline.

Method	Toyota-SM mCA. (↑)	GFlops (↓)
VideoMAEv2-base	73.14	360
+PO-GUISE	76.98	**249**
Internvideo2	75.64	509
+PO-GUISE	**77.03**	399

Table 3: Test results on Toyota-Smarthome (CS) with RGB-only modality at inference.

To demonstrate the flexibility of PO-GUISE and its ability to be integrated into other ViT-based backbones, we have performed an additional experiment using InternVideo2-B/14 [18], see Table 3. It increases the accuracy of VideoMAE by 2.5%, but with 41% more GFLOPS. With this model, the behavior of PO-GUISE is similar. It reduces the number of GFLOPS by a remarkable 27% while increasing the accuracy by 1.5%. In the rest of the paper we use VideoMAEv2-base as the backbone.

4.3.2. Efficiency analysis.

In this experiment we explore the trade-off between accuracy and computational cost incurred by different token selection methods applied on the multi-task model, baseline+HM(P). In Fig. 7 we show the curves of GFLOPS vs. accuracy obtained by training with different values of ρ and λ. For the experiments +HM(P)+PR(C+P) and +HM(P)+PR(P) $\rho \in \{0.3, 0.4, 0.55, 0.7\}$. For the +HM(P)+PR(C+P)+MG experiments, $\rho \in \{0.3, 0.4, 0.45, 0.6\}$ and $\lambda \in \{0.1, 0.2, 0.2, 0.3\}$.

The curve associated with PO-GUISE (baseline+HM(P)+PR(C+P)+MG) is always on top for different proportions of selected tokens (ρ). Interestingly,

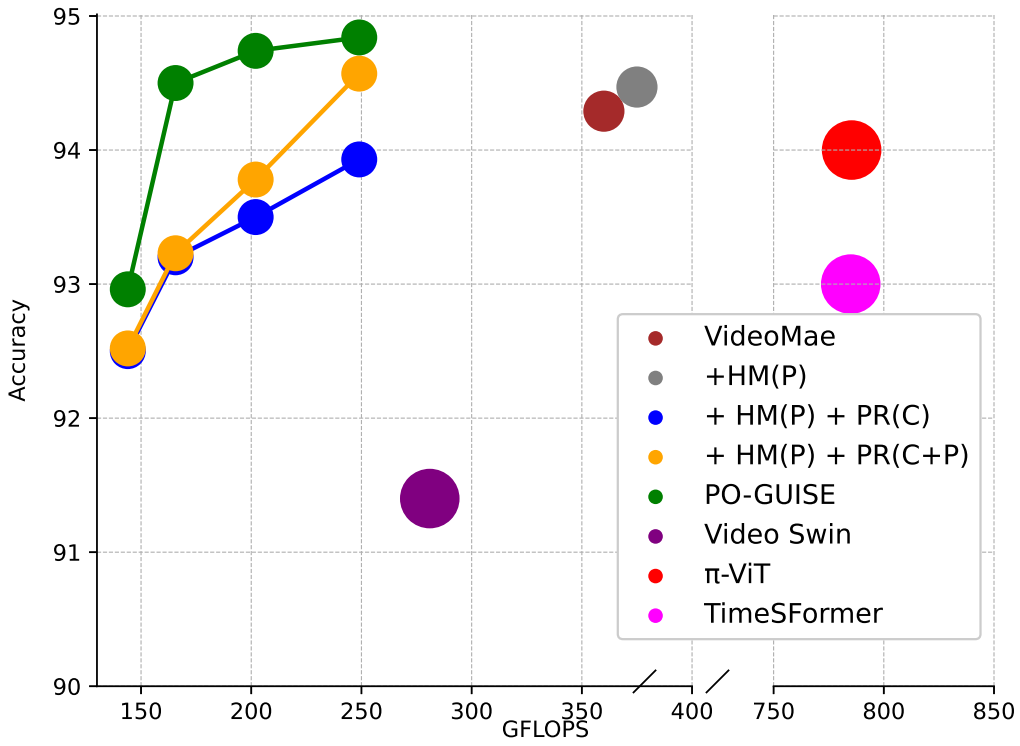


Figure 7: Comparison between GFLOPS and accuracy for different configurations and top methods from SOTA in NTU60 (CS). Circle size represents the number of parameters, either 89M or 121M.

at 166 GFlops our accuracy is still 94.50%, on top of previous methods. The difference with the same pruning method but without token merging (PR(C+P)) is significant, while not using pose tokens in pruning reduces even more the performance in all values of ρ.

We have also conducted experiments on a Jetson Orin NX (16GB) to evaluate performance in a resource-limited device. The baseline model VideoMAEv2 processes one sample every 1140 ms with 3608 MB memory usage. This further increases to 1290 ms, and 4125 MB when incorporating human pose estimation. PO-GUISE at 249 GFLOPS reduces these to 640 ms and 2973 MB, effectively decreasing by 50% and 27% the computational time and cost. This gain in performance is especially important in the Jetson architecture, where the GPU and CPU share the same unified memory, meaning that a lower model memory requirement leaves more space for other secondary CPU tasks. Our memory usage, 2973 MB, also makes it feasible to implement it on the lower-end Jetson models with 4 GB of memory.

4.3.3. Visualizations.

In this section we show some qualitative results at low token keep rates of our improved token selection method PO-GUISE against the top performer token pruning technique [12], Top-K, and the baseline VideoMAEv2-base model. For a fair comparison, we have configured both models to have a similar number of visual tokens and GFLOPS. Specifically, PO-GUISE uses the keep rates $\rho = 0.1, \lambda = 0.1$ and the Top-K model uses $\rho = 0.2$. In Fig. 8 we show some examples, each square represents a visual token and its normalized attention to class token. If a visual token was selected more than once in time, its attention is aggregated. For ease of comparison, we have

used the same color map as in Fig. 1. We can see that PO-GUISE effectively selects the tokens related to the person, while Top-K and the Baseline tend to select irrelevant tokens. We believe this is a side-effect from training ViTs. At inference, these use low-informative background areas of images as a form of repurposed internal computation [48].

The human pose detection task is well learned by the PO-GUISE as shown in Figs. 9 and 10. Note that we are learning one motion heatmap per body joint which consists of the sum of probability maps from the 16 frames of the clip. For ease of visualization, we show in the same image the motion heatmaps corresponding to all body joints.

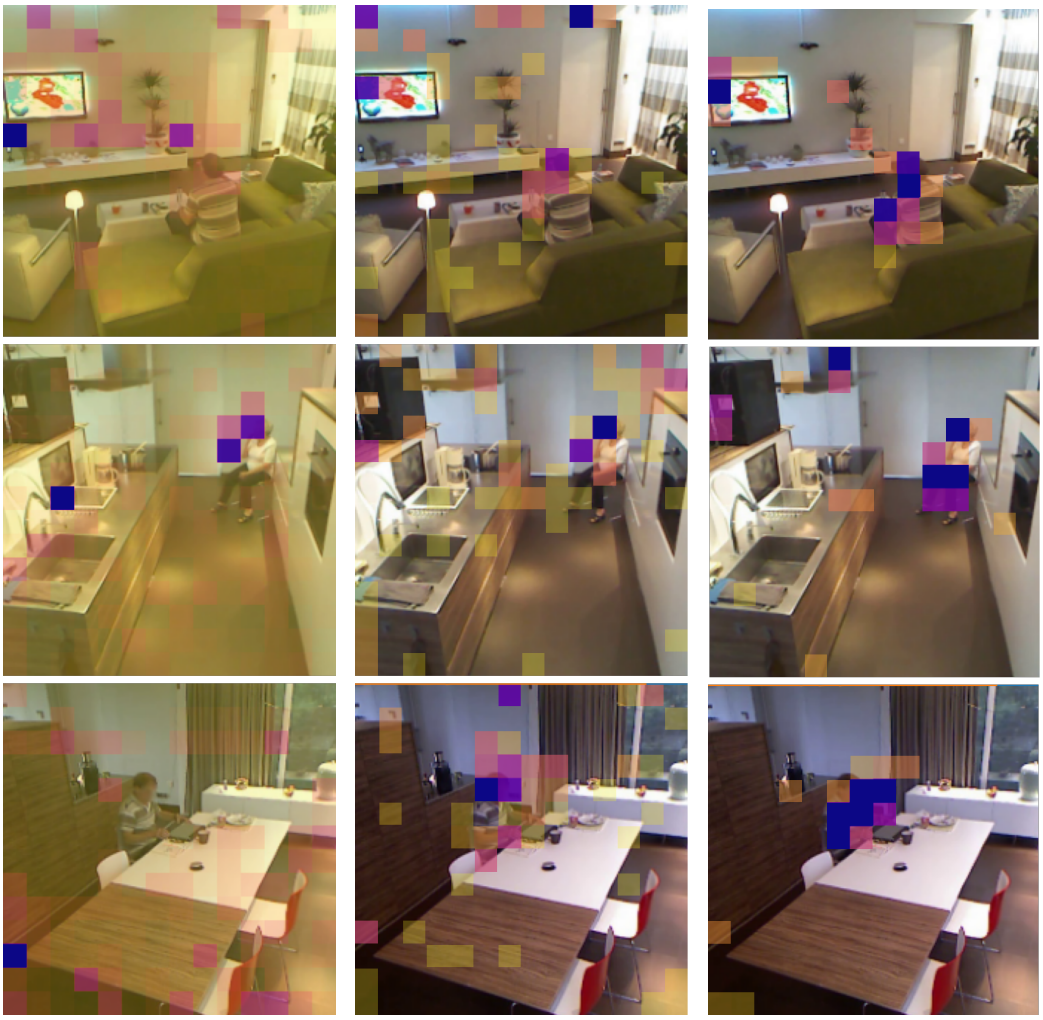
4.3.4. Discussion.

Our contribution is a token selection procedure guided by human motion that, at default settings, not only maintains, but improves the accuracy of a top-performing video transformer. Unlike previous methods, our approach results in a reduction of 30% in GFLOPs. However, since we guide the attention of the transformer towards areas with human motion, it also results in a final increase of the accuracy.

4.4. Comparison with the state-of-the-art

We compare PO-GUISE with state-of-the-art techniques in different ADL recognition datasets: NTU60, NTU120 (Table 5), and Toyota-Smarthome (Table 4).

Our method achieves new state-of-the-art results on the Toyota-SmartHome dataset (Table 4), surpassing the previous state-of-the-art, π-ViT [10], by 4.07%, 3.77%, and 11.32% across all protocols, respectively. The lower per-



(a) Baseline (b) Top-K Pruning (c) PO-GUISE

Figure 8: Visual Token Attention and Selection. Brighter colors indicate higher attention from the selected visual tokens to the class token. For Top-K Pruning and PO-GUISE, we show the attention from the selected tokens at the last stage. For the baseline, the attention maps are obtained from the last layer.

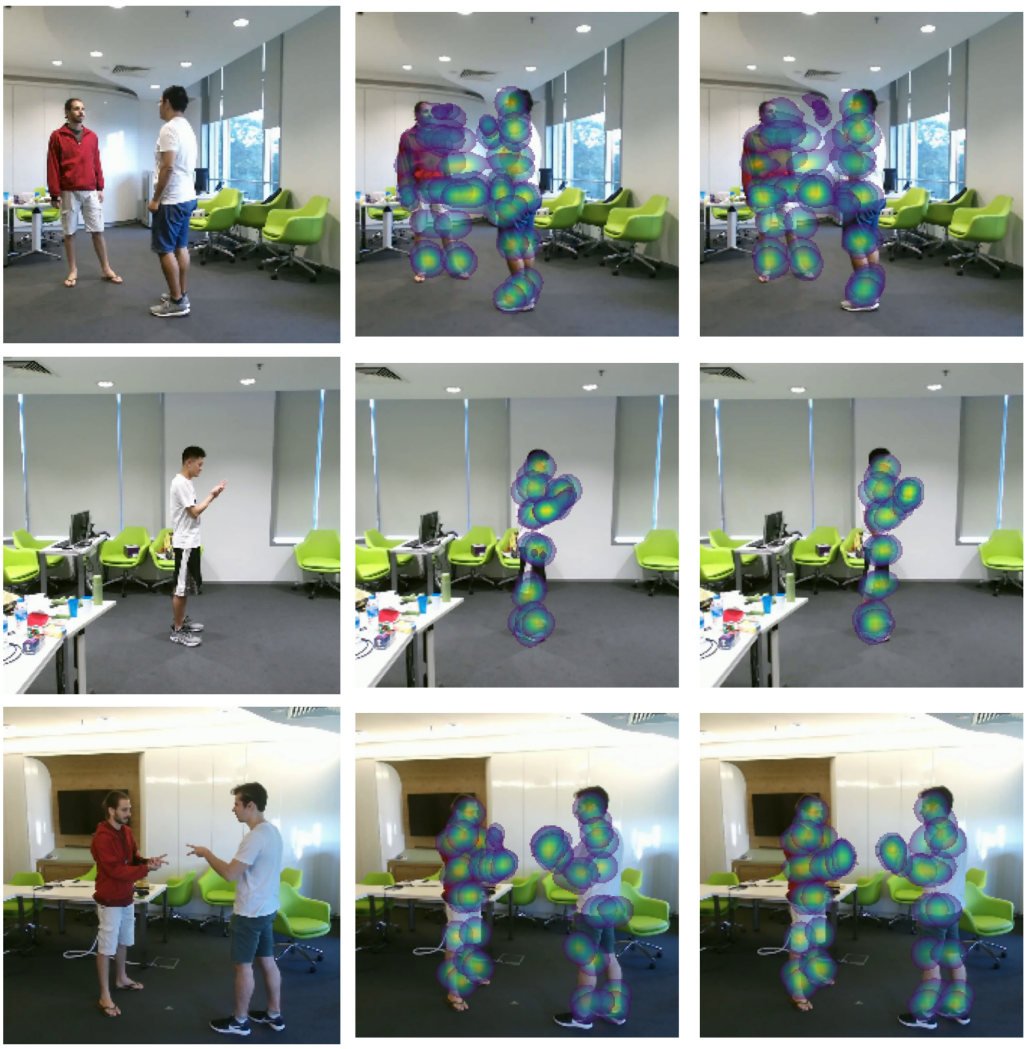


Figure 9: Sample heatmaps from the NTU120 (CS) dataset test set using PO-GUISE. The first column corresponds to the middle frame of the video clip, the second column displays the temporal heatmaps used as training labels, and the third column shows the predicted heatmaps.

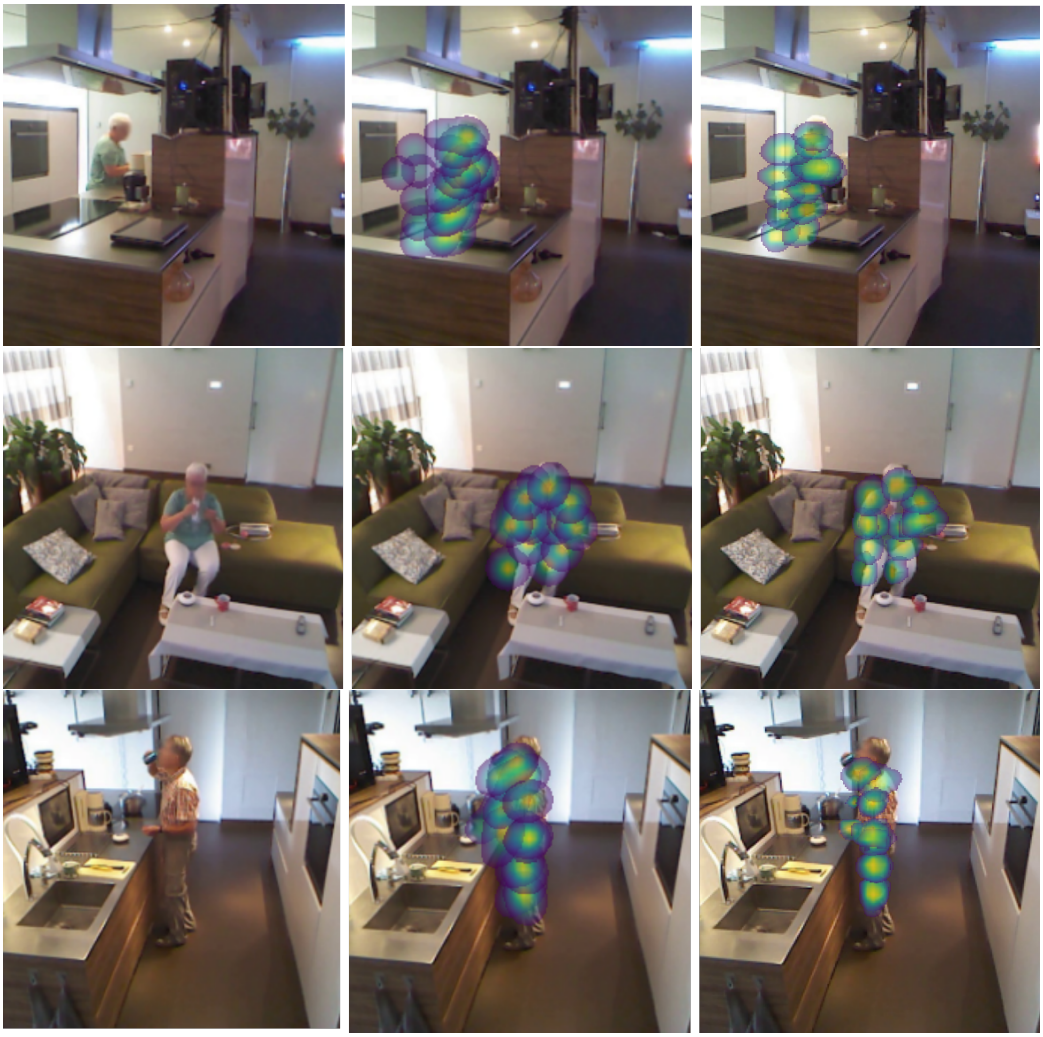


Figure 10: Sample heatmaps from the Toyota-SmartHome (CS) dataset test set using POGUISE. The first column corresponds to the middle frame of the video clip, the second column displays the temporal heatmaps used as training labels, and the third column shows the predicted heatmaps.

Method	CS mCA. (↑)	CV1 mCA. (↑)	CV2 mCA. (↑)	GFlops (↓)
AssembleNet++[49]	63.6	-	-	-
MotionFormer[50]	65.8	45.2	51.0	369
LTN[51]	65.9	-	54.6	-
TimeSFormer [52]	68.4	50.0	60.6	784
VPN++ [1]	69.0	-	54.9	-
Video Swin [34]	69.8	36.6	48.6	281
π-ViT [10]	72.9	55.2	64.8	785
VideoMAEv2-base	73.14	55.20	67.68	360
+ HM(P)	76.01	57.31	71.82	379
PO-GUISE	**76.98**	**58.98**	**76.12**	**249**

Table 4: Test results on Toyota-Smarthome over the CS, CV1 and CV2 protocols.

Method	NTU60		NTU120		GFlops
	CS Acc. (↑)	CV Acc. (↑)	CS Acc. (↑)	CSet Acc. (↑)	(↓)
VideoCon [53]	91.4	98.0	85.6	87.5	-
ViewCLR [54]	89.7	94.1	86.2	84.5	-
VPN++ [1]	93.5	**99.1**	86.7	89.3	-
MotionFormer[50]	85.7	91.6	87.0	87.9	369
TimeSFormer [52]	93.0	97.2	90.6	91.6	784
Video Swin [34]	93.4	96.6	91.4	92.1	281
π-ViT [10]	94.0	97.9	91.9	**92.9**	785
VideoMAEv2-base	94.29	90.91	91.73	89.64	360
+ HM(P)	94.47	91.27	93.36	91.02	379
PO-GUISE	**94.84**	92.31	**93.47**	92.11	**249**

Table 5: Test results on NTU datasets with RGB-only modality at inference.

formance observed in the CV1 protocol, compared to other protocols, is consistent with previous work due to the limited training data available for this challenging single-camera setting.

In the NTU datasets (Table 5), we also surpass state-of-the-art performance on all cross-subject benchmarks compared to methods utilizing only RGB input. PO-GUISE outperforms the prior results of π-ViT [10] by 0.84%, and 1.57% on each dataset's cross-subject protocol (CS), respectively. Importantly, we achieve these performance gains while simultaneously reducing the computational cost of π-ViT by 536 GFLOPS.

The difference in performance observed between the Toyota-SmartHome and NTU datasets for cross-view protocols reflects the difference in difficulty between these benchmarks. In Toyota-SmartHome, the test cameras maintain a similar viewpoint to the training cameras, mostly changing the room the subject is present in. The NTU datasets, and NTU 60 in particular, present a significantly more challenging cross-view scenario, where the cameras used during testing are placed quite differently compared to those utilized for training. However, the difference in size between these datasets explains the better accuracy in NTU. Previous methods have attempted to address this challenge by incorporating 3D pose information during training, π -ViT [10] and VPN++ [1]. Overall, these results highlight the effectiveness of PO-GUISE in cross-subject protocols, with the use of 3D pose information as a promising avenue for future work focused on cross-view protocols.

5. Conclusions

State-of-the-art video transformers for action recognition operate with a quadratic complexity regarding the number of input tokens, which presents a significant computational challenge. Although token pruning offers a promising approach to reduce this computational burden, existing methods often lead to a decrease in action recognition accuracy.

Our method addresses this limitation by leveraging human motion information to selectively retain the most informative tokens for action recognition. This approach achieves a compelling balance between accuracy and computational efficiency. Specifically in default settings, our method reduces the number of visual tokens, resulting in a 30% reduction in GFLOPS while simultaneously increasing accuracy by up to 8%.

Although our method demonstrates notable success on all cross-subject benchmarks, further research is needed to enhance computational efficiency and accuracy on more challenging cross-view action recognition tasks. Our future work will explore the integration of additional semantic tasks to further improve token selection, as well as the incorporation of 3D pose information during training.

The models and code required to reproduce the experiments described in this paper will be made publicly available upon publication.

6. CRediT authorship contribution statement

RP: Software, Investigation, Writing – original draft, Writing – review & editing. RV: Investigation, Methodology, Writing – original draft, Writing – review & editing. JMB: Conceptualization, Methodology, Writing – original

draft, Writing – review & editing, Supervision. LMB, LB: Conceptualization, Methodology, Writing – review & editing, Supervision, Funding Acquisition.

7. Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

8. Data availability

The authors do not have permission to share data, as we are using public datasets owned by other researchers.

9. Funding sources

This work has been supported by the projects PID2021-126623OB-I00, TED2021-130131A-I00, PDC2022-133470-I00, PID2022-137581OB-I00 and by NextGenerationEU/PRTR, PLEC2023-010343 (INARTRANS 4.0) all from MICIU/AEI/10.13039/501100011033/FEDER, UE. RP, JMB, LMB and LB are members of the Madrid ELLIS Unit, funded by the Autonomous Community of Madrid.

References

[1] S. Das, R. Dai, D. Yang, F. Bremond, Vpn++: Rethinking video-pose embeddings for understanding activities of daily living, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (2022) 9703–9717. doi:10.1109/TPAMI.2021.3127885.

[2] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (Eds.), NeurIPS, volume 27, 2014.

[3] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. V. Gool, Temporal segment networks: Towards good practices for deep action recognition, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), ECCV, volume 9912 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 20–36.

[4] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: CVPR, 2017, pp. 4724–4733.

[5] C. Feichtenhofer, A. Pinz, R. P. Wildes, Spatiotemporal residual networks for video action recognition, in: NeurIPS, 2016, pp. 3468–3476.

[6] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A closer look at spatiotemporal convolutions for action recognition, in: CVPR, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 6450–6459.

[7] J. Lin, C. Gan, S. Han, TSM: temporal shift module for efficient video understanding, in: ICCV, IEEE, 2019, pp. 7082–7092.

[8] Y. Chen, D. Chen, R. Liu, H. Li, W. Peng, Video action recognition with attentive semantic units, in: ICCV, IEEE, 2023, pp. 10136–10146.

[9] L. Chen, Z. Tong, Y. Song, G. Wu, L. Wang, Efficient video action detection with token dropout and context refinement, in: ICCV, 2023.

[10] D. Reilly, S. Das, Just add?! pose induced video transformers for understanding activities of daily living, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 18340–18350.

[11] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, Y. Qiao, Videomae v2: Scaling video masked autoencoders with dual masking, in: CVPR, 2023, pp. 14549–14560.

[12] J. B. Haurum, S. Escalera, G. W. Taylor, T. B. Moeslund, Which tokens to use? investigating token reduction in vision transformers, in: IEEE/CVF International Conference on Computer Vision, ICCV 2023 - Workshops, Paris, France, October 2-6, 2023, IEEE, 2023, pp. 773–783. URL: https://doi.org/10.1109/ICCVW60793.2023.00085. doi:10.1109/ICCVW60793.2023.00085.

[13] D. Bolya, C. Fu, X. Dai, P. Zhang, C. Feichtenhofer, J. Hoffman, Token merging: Your vit but faster, in: ICLR, 2023.

[14] Y. Liang, C. Ge, Z. Tong, Y. Song, J. Wang, P. Xie, Evit: Expediting vision transformers via token reorganizations, in: ICLR, 2022.

[15] S. Wei, T. Ye, S. Zhang, Y. Tang, J. Liang, Joint token pruning and squeezing towards more aggressive compression of vision transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 2092–2101.

[16] C. Zhang, Y. Gao, T. Meng, T. Wang, Partitioned token fusion and

pruning strategy for transformer tracking, Image and Vision Computing 154 (2025) 105431.

[17] S. Das, R. Dai, M. Koperski, L. Minciullo, L. Garattoni, F. Bremond, G. Francesca, Toyota smarthome: Real-world activities of daily living, in: The IEEE International Conference on Computer Vision (ICCV), 2019.

[18] Y. Wang, K. Li, X. Li, J. Yu, Y. He, C. Wang, G. Chen, B. Pei, R. Zheng, J. Xu, Z. Wang, et al., Internvideo2: Scaling video foundation models for multimodal video understanding, arXiv preprint arXiv:2403.15377 (2024).

[19] W. Du, Y. Wang, Y. Qiao, RPAN: an end-to-end recurrent pose-attention network for action recognition in videos, in: ICCV, IEEE Computer Society, 2017, pp. 3745–3754.

[20] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, M. J. Black, Towards understanding action recognition, in: ICCV, IEEE Computer Society, 2013, pp. 3192–3199.

[21] V. Choutas, P. Weinzaepfel, J. Revaud, C. Schmid, Potion: Pose motion representation for action recognition, in: CVPR, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 7024–7033.

[22] M. Liu, J. Yuan, Recognizing human actions as the evolution of pose estimation maps, in: CVPR, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 1159–1168.

[23] A. Yan, Y. Wang, Z. Li, Y. Qiao, Pa3d: Pose-action 3d machine for video recognition, in: CVPR, 2019, pp. 7914–7923.

[24] A. Shah, S. Mishra, A. Bansal, J. Chen, R. Chellappa, A. Shrivastava, Pose and joint-aware action recognition, in: IEEE Winter Conf. on Appl. of Comput. Vis., IEEE, 2022, pp. 141–151.

[25] D. Ahn, S. Kim, H. Hong, B. C. Ko, Star-transformer: A spatio-temporal cross attention transformer for human action recognition, in: IEEE Winter Conf. on Appl. of Comput. Vis., 2023, pp. 3330–3339.

[26] S. Kim, D. Ahn, B. C. Ko, Cross-modal learning with 3d deformable attention for action recognition, in: ICCV, 2023, pp. 10231–10241.

[27] H. Zhang, M. C. Leong, L. Li, W. Lin, Pgvt: Pose-guided video transformer for fine-grained action recognition, in: IEEE Winter Conf. on Appl. of Comput. Vis., 2024, pp. 6645–6656.

[28] H. Duan, Y. Zhao, K. Chen, D. Lin, B. Dai, Revisiting skeleton-based action recognition, in: CVPR, 2022, pp. 2959–2968.

[29] A. Holzbock, A. Tsaregorodtsev, Y. Dawoud, K. Dietmayer, V. Belagiannis, A spatio-temporal multilayer perceptron for gesture recognition, in: 2022 IEEE Intelligent Vehicles Symposium (IV), 2022, pp. 1099–1106. doi:10.1109/IV51971.2022.9827054.

[30] Z. Li, H. Guo, L.-P. Chau, C. H. Tan, X. Ma, D. Lin, K.-H. Yap, Object-augmented skeleton-based action recognition, in: 2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS), 2023, pp. 1–4.

[31] M. Martin, D. Lerch, M. Voit, Viewpoint invariant 3d driver body pose-based activity recognition, in: IEEE Intelligent Vehicles Symposium, IV, IEEE, 2023, pp. 1–6.

[32] D. C. Luvizon, D. Picard, H. Tabia, 2d/3d pose estimation and action recognition using multitask deep learning, in: CVPR, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 5137–5146.

[33] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, C. Schmid, Vivit: A video vision transformer, in: ICCV, IEEE, 2021, pp. 6816–6826.

[34] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, H. Hu, Video swin transformer, in: CVPR, IEEE, 2022, pp. 3192–3201.

[35] H. Ma, Z. Wang, Y. Chen, D. Kong, L. Chen, X. Liu, X. Yan, H. Tang, X. Xie, Ppt: token-pruned pose transformer for monocular and multi-view human pose estimation, in: ECCV, Springer, 2022, pp. 424–442.

[36] S. Kim, G.-J. Yoon, J. Song, S. M. Yoon, Simultaneous image patch attention and pruning for patch selective transformer, Image and Vision Computing 150 (2024) 105239.

[37] Z. Wang, X. Lin, N. Wu, L. Yu, K.-T. Cheng, Z. Yan, Dtmformer: Dynamic token merging for boosting transformer-based medical image segmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 2024, pp. 5814–5822.

[38] Y. Xu, J. Zhang, Q. Zhang, D. Tao, Vitpose: Simple vision transformer baselines for human pose estimation, Advances in Neural Information Processing Systems 35 (2022) 38571–38584.

[39] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, A. C. Kot, Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding, IEEE transactions on pattern analysis and machine intelligence 42 (2019) 2684–2701.

[40] A. Navon, A. Shamsian, I. Achituve, H. Maron, K. Kawaguchi, G. Chechik, E. Fetaya, Multi-task learning as a bargaining game, arXiv preprint arXiv:2202.01017 (2022).

[41] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[42] I. Loshchilov, F. Hutter, SGDR: Stochastic gradient descent with warm restarts, in: International Conference on Learning Representations, 2017. URL: https://openreview.net/forum?id=Skq89Scxx.

[43] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6023–6032.

[44] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018. URL: https://openreview.net/forum?id=r1Ddp1-Rb.

[45] E. D. Cubuk, B. Zoph, J. Shlens, Q. V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in: Pro-

ceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 702–703.

[46] J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, M. Suo, P. Tillet, E. Wang, X. Wang, W. Wen, S. Zhang, X. Zhao, K. Zhou, R. Zou, A. Mathews, G. Chanan, P. Wu, S. Chintala, PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation, in: 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24), ACM, 2024. URL: https://pytorch.org/assets/pytorch2-2.pdf.

[47] L. Biewald, Experiment tracking with weights and biases, 2020. URL: https://www.wandb.com/, software available from wandb.com.

[48] T. Darcet, M. Oquab, J. Mairal, P. Bojanowski, Vision transformers need registers, in: The Twelfth International Conference on Learning Representations, 2024.

[49] M. S. Ryoo, A. Piergiovanni, J. Kangaspunta, A. Angelova, Assemblenet++: Assembling modality representations via attention connections, in: Computer Vision–ECCV 2020: 16th European Conference,

Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16, Springer, 2020, pp. 654–671.

[50] M. Patrick, D. Campbell, Y. Asano, I. Misra, F. Metze, C. Feichtenhofer, A. Vedaldi, J. F. Henriques, Keeping your eye on the ball: Trajectory attention in video transformers, Advances in neural information processing systems 34 (2021) 12493–12506.

[51] D. Yang, Y. Wang, Q. Kong, A. Dantcheva, L. Garattoni, G. Francesca, F. Brémond, Self-supervised video representation learning via latent time navigation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 3118–3126.

[52] G. Bertasius, H. Wang, L. Torresani, Is space-time attention all you need for video understanding?, in: Proceedings of the International Conference on Machine Learning (ICML), 2021.

[53] K. Shah, A. Shah, C. P. Lau, C. M. de Melo, R. Chellapp, Multi-view action recognition using contrastive learning, in: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2023, pp. 3370–3380. doi:10.1109/WACV56688.2023.00338.

[54] S. Das, M. S. Ryoo, Viewclr: Learning self-supervised video representation for unseen viewpoints, in: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2023, pp. 5562–5572. doi:10.1109/WACV56688.2023.00553.