OO-dMVMT: A Deep Multi-view Multi-task Classification Framework for Real-time 3D Hand Gesture Classification and Segmentation

Federico Cunico* Federico Girella* Andrea Avogaro*
Marco Emporio* Andrea Giachetti Marco Cristani

University of Verona

name.surname@univr.it

Abstract

Continuous mid-air hand gesture recognition based on captured hand pose streams is fundamental for human-computer interaction, particularly in AR / VR. However, many of the methods proposed to recognize heterogeneous hand gestures are tested only on the classification task, and the real-time low-latency gesture segmentation in a continuous stream is not well addressed in the literature. For this task, we propose the On-Off deep Multi-View Multi-Task paradigm (OO-dMVMT). The idea is to exploit multiple time-local views related to hand pose and movement to generate rich gesture descriptions, along with using heterogeneous tasks to achieve high accuracy. OO-dMVMT extends the classical MVMT paradigm, where all of the multiple tasks have to be active at each time, by allowing specific tasks to switch on/off depending on whether they can apply to the input. We show that OO-dMVMT defines the new SotA on continuous/online 3D skeleton-based gesture recognition in terms of gesture classification accuracy, segmentation accuracy, false positives, and decision latency while maintaining real-time operation.

1. Introduction

The current generation of Mixed Reality (MR) headsets (e.g. Meta Quest 2, Hololens 2, etc.) features accurate hand tracking capabilities, capturing finger poses at high frame rates, and cheap sensors/APIs are available to enhance different applications with this data. Real-time, reliable recognition of mid-air hand gestures is a fundamental ingredient for building 3D "natural" interfaces for MR, Human-Robot, and Human-Machine interaction in industry, home appliance control, and more.

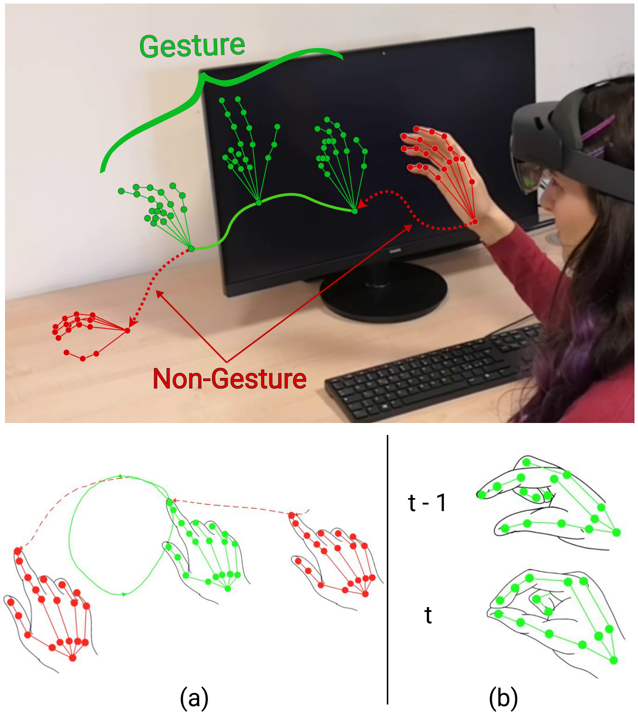

Figure 1. On top, the 3D hand gesture real-time classification and segmentation scenario: in red are non-gestures, in green the segmented gesture. The task is difficult: below, we show samples from SHREC'19 (a), where it is possible to see long gestures with a static pose of the hand and the correct start of the gesture cannot be easily detected; in (b) a sample form SHREC'22, where gestures express more pose dynamics.

In these contexts, the gestures have to be detected and classified within a continuous stream of hand movements that can be captured by cameras or other sensors. A variety of skeleton-based architectures for hand gesture recognition have been proposed over the years, leveraging different methodologies such as multi-view learning [11, 29, 31], attentive models [24], graph convolutional networks [3, 8, 18, 19, 21, 22, 25, 27], LSTMs [1]. Most of these methods use features extracted from the data, proving the importance

*The authors contributed equally to this paper
Code: https://github.com/intelligolabs/OO-dMVMT

of enriching the input, rather than using all the raw data.

However, none of the methods proposed in the literature has the accuracy and a false positives' score good enough to guarantee a natural and reliable MR interaction. A missed or wrong gesture classification can potentially be dangerous for the users if they have to interact with the machinery of an industry, medical operation, or any delicate situation.

In order to build an effective model, we tackled the online gesture recognition problem using a powerful yet unexplored framework called Multi-View Multi-Task (MVMT) learning paradigm [17, 23]. In MVMT, complementary views of the object of interest serve to learn a rich feature embedding, that is fed into multiple tasks to borrow information across tasks that can potentially increase the predictive power of the learned models. Multi-View (MV) learning [33] is a well-established direction in machine learning which associates multiple feature sets (called "views") to the input data *e.g.* given an image, extract spatial features, color features and frequency features to form a single data point. Each data point is thus associated with V sets of different features where each set corresponds to a view, allowing supervised or semi-supervised learning tasks to access a more complete data descriptor. MV learning does not have to be confused with the multi-camera approach in which multiple camera views create their data point, that are then usually merged (synchronized) into a single one. Multi-Task (MT) learning aims to leverage useful information contained in multiple related tasks to help improve the generalization performance of at least one of the tasks [32].

MVMT learning is a less explored field, which extends MV learning to the MT learning setting with T tasks, where each task is a multi-view learning problem [17]. Most of the existing MVMT methods focus on proposing linear models for fitting the specific application requirements even though they are not suitable for common large-scale real-world problems in real environments. Only recently, non-linear deep models have been taken into account [30]. In our case, we take into account multiple views related to hand geometry and its global movement to create a multi-view description. Furthermore, we raise the accuracy by exploiting the relatedness of multiple tasks such as gesture classification at different grains and regression of its start-end instants.

The issue arises when some tasks can't be performed under certain conditions, such as the regression of the start frame of a gesture when no gestures are active. This invalidates the classical MVMT framework, which requires all the tasks to be active at each time. For this reason, we propose the On-Off deep Multi-View Multi-Task paradigm (OO-dMVMT), which extends MVMT by allowing specific tasks to switch on/off depending on whether they can process the observation sample.

OO-dMVMT deals exactly with this situation, introducing a mechanism that, at training time, switches tasks on and off depending on whether they can act on a specific data observation. Our contributions are the following:

- We individuate MVMT as the correct framework to deal with real-time hand gesture recognition.
- We extend MVMT to adapt to cases when some of the tasks cannot operate on a specific data sample, introducing OO-dMVMT
- With OO-dMVMT, we define the new state-of-the-art of hand gestures real-time classification and segmentation.

We show the performance of our method in the realistic scenario proposed by specific benchmarks (SHREC'19 [5], SHREC'22 [13]), where gestures live within long sequences of non-significant hand movements from which they need to be distinguished .

2. Related works

Hand gestures designed for AR/VR interaction have short but variable lengths, varying action granularity, and their semantics may depend on completely different factors (the static pose, the global hand trajectory, the fingers' articulation). While many classifiers have been proposed to recognize gestures of this kind from hand poses' sequences, most of them were not tested in a realistic scenario, but only tested on offline benchmarks like DHG14/28 [9] and SHREC'17 [10]. These benchmarks only evaluate the labeling accuracy of pre-segmented gestures obtained with classifiers trained on similar data.

Offline classification methods. For this task different network architectures have been proposed. Devineau et al. [11] proposed a multi-channel convolutional neural network with two feature extraction modules and a residual branch per channel. Maghoumi et al. [24] proposed DeepGRU, based on a set of stacked gated recurrent units (GRU) and a global attention model. Hou et al. [18] employed an end-to-end Spatial-Temporal Attention Residual Temporal Convolutional Network (STA-Res-TCN). Spatial-temporal GCN has been also adapted for hand gesture recognition [21]. Chen et al. [8] built a fully connected graph from the hand skeleton and learned node features and edges via a self-attention mechanism that performs in both spatial and temporal domains. Liu et al. [22] extracted optimized features from skeleton data by using a disentangled multi-scale aggregation scheme. Li et al. [19] proposed a two-stream network to address the variable temporal scales of the gesture classes. The first stream extracts short-term temporal information and hierarchical spatial information, and the second one extracts long-term temporal information. Shi et al. [27] modeled the spatial-temporal dependencies between joints without the requirement of knowing their positions or mutual connections by employing a Spatial-Temporal Attention Network.

A key observation looking at the literature on segmented gesture classification is that, while there is no evidence of particular advantages in the accuracy obtained with different network architectures (recurrent, graph-convolutional, 1D convolutional networks), it seems that better classification performances have been obtained by feeding the networks with pre-computed features instead of raw data, often organized in multiple sets with different semantics. Avola et al. [1] used a stack of LSTM units fed with angles at fingers' joints, intra-finger angles, fingertips', and hand center displacements. In [7] global and local motion features, along with the skeleton sequence, are fed into different branches of a Long-Short Term Memory (LSTM) network to get the predicted class of input gesture. Yang et al. [31] proposed DDNet, a simple network based on 1D convolutions, fed with multiple features (linearized joints' distance matrix, joints speeds) with a motion summarising module to reduce noise from non-relevant frames. Trivedi et al. [29] used multiple features (joints, bones, joints' velocity, bones velocity) to feed a Spatio-Temporal Relational Module. The use of the pre-computed features results in lighter and easier to train networks and the parallel encoding of different sets of features can be interpreted as an application **multi-view** learning paradigm. Note that multi-view here does not mean that multiple features are related to different physical viewpoints (even if this idea has been proposed as well, for example, in [20]).

Continuous (online) classification. To build effective interfaces, we need to perform continuous or "online" recognition, which requires performing multiple tasks: localizing the gestures in the pose stream and labeling them by only using past information, providing results with a small delay, and avoiding detection not corresponding to actual gestures (false positives). Specific benchmarks have only recently been proposed. The first is SHREC'19 [5], that tests continuous recognition, but features only simple dynamic gestures characterized by global trajectories without hand articulation (see Sec. 4.1). The benchmark we adopted to train and evaluate our system is SHREC'22 [13] where the gesture dictionary is heterogeneous, including static and dynamic gestures similar to those proposed in DHG14/28 and SHREC'17 (see Sec. 4.1). This benchmark improves the previous SHREC'21 [4], proposed by the same authors and with a similar structure, but solving some issues that, according to the website, made it unreliable for comparative tests. For these benchmarks, training data are long sequences of pose streams with annotated start frames, end frames, and labels of the gestures executed and the task is to detect the occurrences of gestures in test sequences using past information only. SHREC'22 data are captured with an Hololens 2 headset, *meaning that a recognizer trained for the task could be directly integrated into an interactive demo* using the same device to capture hand poses.

Two classes of approaches can be adopted to handle the online task [14]: *direct* and *indirect* methods. Direct methods employ specialized heuristics based on speed, energy, or curvature (or trained networks) to find gesture boundaries and then send the extracted candidates to a classification module. Although this approach is not expected to work well in the case of complex motions/gestures [28], it has been proposed in different implementations. In SHREC'19 [5], Seg.LSTM1 uses an LSTM together with a specialized segmentation network. The ST-GCN method used in [4] uses an energy-based segmentation approach adding several ad-hoc rules. In the 2ST-GCN method of [13], an energy-based detection module is combined with a fine-grained classifier, providing gesture/non-gesture discrimination. Indirect methods perform a continuous classification (simultaneous detection and labeling). This can be done using pre-trained classifiers working with fixed-size input sub-sequences and sliding windows schemes or by using recurrent networks like Long-Short Time Memory (LSTM) networks or Gated Recurrent Units (GRU). A simplified version of DeepGRU [24], combined with a smart data augmentation method and adapted for online detection, performed best in the SHREC'19 contest [5], but a similar solution did not perform well on complex heterogeneous gestures [4]. The use of sliding windows is a simple and popular solution for the online task. Two main ideas have been proposed in this context: train the classifier on segmented gestures, handling the unknown input gesture length in the online testing [12], or train it on fixed-size sub-parts (solution adopted by two methods presented in [13]). In [12] a modified DDNet [31] is trained with resampled, segmented gestures and randomly sampled non-gesture windows. The online testing is performed with windows of variable width and the results are combined with a voting procedure. The TN-FSM method proposed in SHREC'22 [13] trains a Transformer Network to classify windows of 10 frames on subsets of the training sequences and uses a Finite State Machine to create the online prediction. Causal TCN, the method providing the best accuracy in [13], trains a temporal convolutional network on 20-frames windows labeled with gesture classes or non-gestures depending on their intersections with the annotated ground truth of the training set. We decided to apply a similar protocol for our online training framework and we tested many of the models proposed for the offline setting in the continuous task, training them with fixed-size windows and adding a non-gesture class. Within this framework, we finally developed our novel On-Off deep Multi-View Multi-Task (OO-dMVMT) method providing enhanced performances with respect to the SotA classifiers. We show that with OO-dMVMT using a sliding window approach we are able to perform not only real-time gesture classification but also gesture segmentation with SotA performances.

3. Our pipeline

We present our proposed OO-dMVMT by first detailing the views and the tasks we have considered (Sec. 3.1 and Sec. 3.2, respectively). Then, we present how to train the model (Sec. 3.3) and how to infer from it for the real-time hand classification problem (Sec. 3.4). To fix the notation, ξ_t is the observation window that ends at frame t of length W, covering the frame interval $[t - W + 1, t]$.

3.1. The views

Every observation training window ξ_t is associated to $V = 3$ views: 1) *Geometric layout*. We borrow from [31] the flattened Joint Collection Distances (JCD) features, which are location-viewpoint invariant. JCD computes the Euclidean distances between a pair of collective joints, and is of size $\binom{J}{2}$ for each frame of the observation window, resulting in a tensor of $\binom{J}{2} \times W$. 2) *Short-term slow motion M_{slow}*. M_{slow} computes the 1-frame linear velocity of every single joint for all the joints, resulting in a tensor of size $J \times W - 1$. 3) *Short-term fast motion M_{fast}*. M_{fast} is similar to the short-term slow motion, but the linear velocity is computed every other frame (skipping the ones in between), with a $J \times \frac{W}{2} - 1$ resulting tensor. In practice, M_{slow} and M_{fast} model the short-term global motion of the skeleton in terms of speed. Following the multi-view learning paradigm, the three views JCD, M_{slow} and M_{fast} are each embedded into a $\mathbb{R}^{\frac{W}{2} \times 8}$ dimensional space by a set of independent encoders and concatenated in a multi-view latent pattern g_t of $\mathbb{R}^{\frac{W}{2} \times 24}$ dimensions.

3.2. The tasks

$T = 4$ tasks have been considered: 1) *SDN classification*. We classify gestures into three main categories: Static gestures (S), Dynamic gestures (D), and Non-Gestures (N). This macro classification is typical in heterogeneous hand gesture modeling [4,13], and every gesture can fall in one of these superclasses. Static gestures require the user to fix the hand in a predetermined pose, keeping it still for a while. Dynamic gestures require the user to move the hand centroid, following a specific trajectory. Non-gestures are all those natural movements of the hand that occur between gestures and are the most difficult class to capture. 2) *Fine-grained classification of gestures*. This task implements the fine-grained classification of the window ξ_t considering all the L classes into play (including the non-gesture class). This is the only task that is activated during testing time. 3) and 4) *Start/end gesture frame regression*. These two tasks perform the regression on the frame indices t_{start}, $t_{end} \in [0, W - 1]$ of the start and end of a gesture, respectively. The idea is to identify the patterns of the start and/or end of a gesture, thus aiding the classification tasks.

These last two regression tasks could be unable to function, due to the presence of observation windows which are completely contained in gesture (or non-gesture) streams. This configuration, dealing with multi-views data and with some tasks which cannot be associated with some (or all) of the views of the input data, is novel in the MVMT literature.

3.3. Model training

Let ξ_t be an input data instance sampled at time t, and k the index of the k-th task. We define the aggregated OO-dMVMT objective function to minimize as:

$$F(\xi_t) = \sum_{k=1}^{T} c(k,t) \cdot F^{(k)}(g_t, w_g, w_k) \qquad (1)$$

$F^{(k)}$ is the deep objective function associated with task k controlled by $c(k,t)$, a boolean selector which is 1 if task k can work with input ξ_t, otherwise ξ_t does not contribute to the weights update for the task k. The decision of whether the task can work with the input is done at train time and depends on the semantics of ξ_t and the task itself. The term g_t represents the multi-view representation of ξ_t (see Sec. 3.1), w_g are the multi-view parameters which generate g_t starting from ξ_t, and w_k are the k-th task parameters that generate the output for the k-th head. The deep objective functions $F^{(k)}$ associated with our tasks are MSE for the regression tasks ($k = 3, 4$) and cross-entropy for the classification tasks ($k = 1, 2$). All the objective functions are uniformly weighted.

The idea is to treat the OO-dMVMT model as an instance of asynchronous optimization [2], where the central server begins to update the shared model w_g after it receives a gradient computation from one task node, without waiting for the other task nodes to finish their computations. In our case, the switching off of a task k corresponds to having a delay for that task, while the other active tasks contribute to the model update.

3.4. Model inference

The pipeline of the framework *at inference time* is summarized in Fig. 2 and consists of three steps: in step 1) the observation window ξ_t is sampled at frame t, and the multi-view description is extracted; in step 2) g_t flows into the fine-grained classification task, and a preliminary classification label $\tilde{y}_t$ is associated with the observation window ξ_t. The other three tasks are considered only during the training stage, as discussed in Sec. 3.3. 3) a post-processing step is carried out, considering the previous $W - 1$ preliminary classifications, in order to remove spurious ones. It is implemented by looking backward at W frames, assigning the class label $y_t = l$ to the observation window ξ_t if $l \in \{0, \ldots, L\}$ is the most frequent label in the last W preliminary classification labels $\tilde{y}_t$. Step 3 ensures a more stable output, as the single window classifications may be

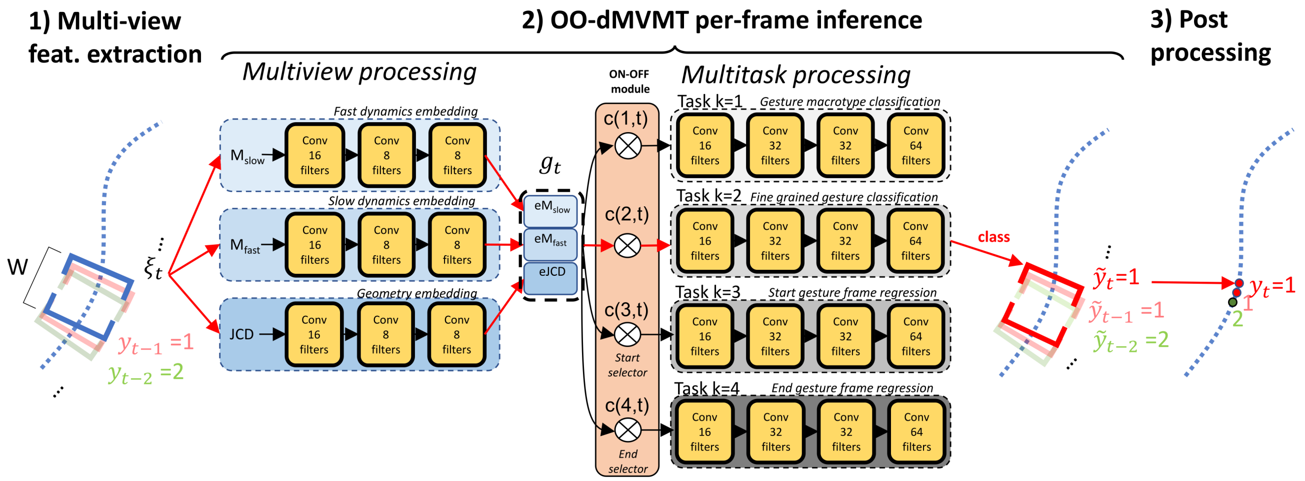

Figure 2. The pipeline of our framework, is composed of three steps: 1) Multi-view feature extraction, 2) OO-dMVMT per-frame inference, 3) Post-processing. In red is highlighted the path used during the testing time, after the OO-dMVMT model has been trained.

noisy if the window is small and the gesture dictionary includes complex examples with similar subparts. It is worth noting that W is an important parameter of the method that needs to be tuned based on the specific characteristics of the gestures' dictionary: it needs to be long enough to capture long gestures, but as short as possible to reduce input delay.

4. Experiments

The experiments are organized as follows: in Sec. 4.1, we detail the popular SHREC'22 [13] and SHREC'19 [5] datasets, together with the associated gesture classification metrics; in Sec. 4.2 we report the results on the SHREC'22 benchmark; in Sec. 4.3 we focus on SHREC'19; finally, we present multiple ablative studies in Sec. 4.4. As comparative approaches, we consider several graph-based architectures like SeS-GCN [25] and MS-G3D [22], the attention-based networks DG-STA [8] and DSTA [27], a 1D convolutional network specifically designed for hand gestures, the Double-feature Double-motion Network (DDNet) [31], and a lightweight action recognition network (PSUMNET [29]). These approaches add to the competitors which can be found in the SHREC'22 challenge [13] and SHREC'19 challenge [5], for a total of 13 comparative methods.

4.1. Datasets and evaluation metrics

SHREC'22. The SHREC'22 benchmark [13] features continuous recordings of 3D hand poses captured in simulated Mixed Reality interactions with an Hololens 2 device. The training set includes 144 sequences, with a total of 36 occurrences for each of the 16 gesture classes, interleaved with non-significant hand movements (non-gestures). The testing set has the same cardinalities. Gestures belong to four categories: static, dynamic coarse, dynamic fine, and periodic. Each sequence is annotated with start frames, end frames, and gesture labels. Training and testing sequences have been captured by different subjects. The benchmark evaluation protocol requires that the recognizer outputs sequences' annotations with the list of recognized gestures, their labels, and the predicted start and end frames. Since online prediction uses time samples after the estimated start, storing this delay is necessary for evaluating interface response time. We follow the protocol and make use of the official evaluation code provided in the contest's repository [13]. The metrics are the following:

Jaccard Index (JI): the average relative overlap between the ground truth and the predicted labels for the input sequences. It is used in many continuous classification tasks, but it does not evaluate the ability to avoid multiple activations for a single gesture or small noisy activations.

Detection rate (DR): the ratio between the number of correctly detected gestures and the total number of gestures in the input sequences. A gesture is considered correctly detected if it has a temporal intersection (referred to as Minimum Overlap Ratio) with the ground truth greater than 50% of the true interval, does not last more than twice the real duration, and has the same label. The gestures predicted by the recognizer but not corresponding to ground truth ones are defined as false positives.

False positive score (FP): defined as the ratio between the number of false positives and the total number of gestures.

Minimal detection delay (Delay): the delay metric is the difference in frames between the gesture start and the last frame used for the prediction.

SHREC'19. The SHREC'19 benchmark [5] is focused on dynamic coarse gestures, characterized by the trajectory of the joints following specific 2D patterns (V-mark, X-mark, Caret, Square, Circle). Hand trajectories have been captured with a Leap Motion sensor on users performing simulated interactive tasks in VR. The dataset is composed of 195 sequences, each one containing a single gesture surrounded by non-meaningful movements labeled as non-gestures. Each sequence includes the 3D coordinates of the hand joints and the quaternions defining bones' orientation.

The evaluation method provided in this benchmark considers an inference as correct if the correct predicted gesture is within 2.5 seconds from the ground truth gesture time window. The metrics used for the evaluation of this benchmark are the DR, FP (defined as for SHREC'22), and the inference time, intended as the time needed to perform a model inference for labeling a single frame.

It is worth noting that SHREC'22 and SHREC'19 have similar evaluation , but quite different dictionaries (heterogeneous types and duration of the classes in the first case, homogeneous in the second) and characteristics (acquisition devices, hand skeleton model, average gesture length).

4.1.1 Implementation details

The architecture of the network is shown in Fig. 2. The view encoders are independent series of 1D Conv layers. Each task branch is a series of 1D Conv layers except for the last layers, which are fully-connected layers (with one final neuron for regression, three for SDN, and L for the fine-grained classifier). We adopted the Adafactor [26] with an initial LR of 0.004 over 100 epochs on 2 NVIDIA RTX3090.

4.1.2 Real-time 3D hand pose

As far as we know, the sequences in the benchmarks have been acquired with off-the-shelf devices and there are no guarantees on the accuracy of the tracking. This can possibly be one of the reasons for the non-optimal results of many methods, and OO-dMVMT achieving better scores shows that it may be more robust against tracking errors. Improvements in hand tracking systems [15, 16] could further reduce the errors without changing the algorithm.

4.2. SHREC'22 results

In Tab. 1 we report the official gesture classification metrics as provided by the evaluation code. For each comparative model, we look for its best parameters. The W parameter should be as small as possible, to lower input delay, but large enough to capture significant information for longer gestures. In our case, we found $W = 16$ to be an optimal choice for SHREC'22.

OO-dMVMT defines the new SOTA in all the metrics, with a fast inference time (4.1ms), and very low variances. Fig. 3 highlights the detection rate per gesture type of the five best performing models on SHREC'22. Notably, OO-dMVMT is the most consistent top performer on this benchmark, with a detection rate that spans between 0.82 and 1, which is the thinnest min-max score range.

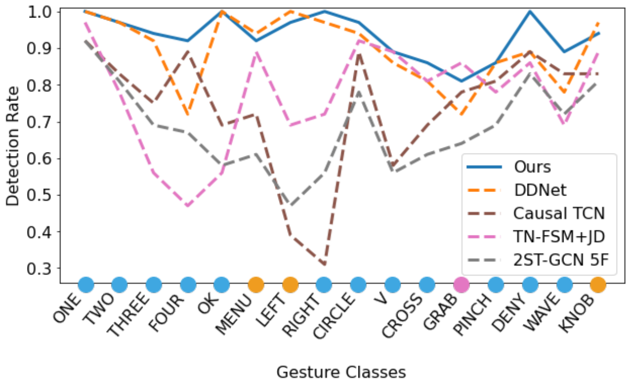

Figure 3. Detection rate per gesture class for SHREC'22. Points are connected with a line for clarity. On the x-axis, a colored dot indicates, for each class, the method that achieved the best detection rate. Only the best 5 methods by DR are shown for clarity.

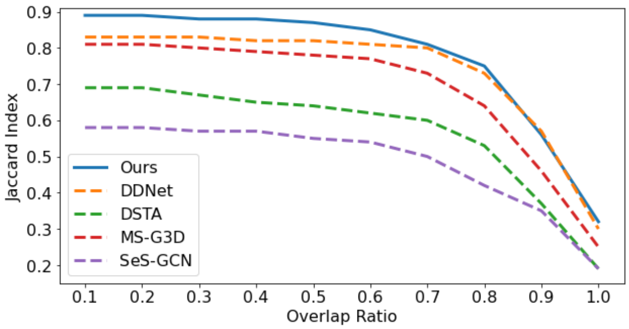

Figure 4. Jaccard index for SHREC'22 as a function of the minimum overlap ratio. For clarity, we show only the best 5 methods.

An important parameter of the evaluation metrics for SHREC'22 [13] is the *minimum overlap ratio* (MOR) of the detected gestures with the ground truth. When MOR $=$ 1 the predicted gesture needs to completely envelop the ground truth, while a value closer to MOR $= 0$ relaxes this constraint. Like in the original evaluation protocol, the results are evaluated with MOR $= 0.5$. Fig. 4 shows the Jaccard Index (described in Sec. 4.1) as a function of the Minimum Overlap Ratio. Obviously, the higher the MOR, the lower the JI of all the approaches. Until MOR $= 0.7$, OO-dMVMT clearly dominates the competitors.

Of particular importance is the FP results, as they would trigger unwanted actions making the interaction frustrating. In Fig. 5, a stacked barplot showing the total FPs for each model is presented (lower is better). Interestingly, the macro category which in general attracts the most false positives is the dynamic one, probably due to the fact that a movement of the limb is necessary to start this kind of gesture. As soon as we move towards more effective approaches, it is easy to

Table 1. Classification results of SHREC'22 benchmark. The metrics are the detection rate (DR), false positives scores (FP), the Jaccard Index (JI), the delay in frames, and per-frame processing time. In brackets are the standard deviations. Columns SV/MV refers to the single-view or multi-view learning approach, while ST/MT refers to single-task or multi-task learning approach.

Method	**DR** ↑	**FP** ↓	**JI** ↑	**Delay(fr.)** ↓	**time(ms)** ↓	SV	MV	ST	MT
DeepGRU [24] (2018)	0.26 *(.14)*	0.25 *(.23)*	0.21 *(.09)*	8.00	3.1	✓		✓	
DG-STA [8] (2019)	0.51 *(.10)*	0.32 *(.20)*	0.40 *(.20)*	8.00	4.2	✓		✓	
SeS-GCN [25] (2022)	0.60 *(.13)*	0.16 *(.09)*	0.53 *(.13)*	8.00	1.8	✓		✓	
PSUMNET [29] (2022)	0.62 *(.14)*	0.24 *(.15)*	0.52 *(.15)*	8.00	24.4		✓	✓	
MS-G3D [22] (2020)	0.68 *(.11)*	0.21 *(.15)*	0.57 *(.14)*	8.00	29.3	✓		✓	
Stronger [13] (2022)	0.72 *(.11)*	0.34 *(.26)*	0.59 *(.18)*	14.79	100.0		✓	✓	
DSTA [27] (2020)	0.73 *(.07)*	0.24 *(.13)*	0.61 *(.12)*	8.00	9.2	✓		✓	
2ST-GCN 5F [13] (2022)	0.74 *(.12)*	0.23 *(.05)*	0.61 *(.11)*	13.28	2.1	✓			✓
TN-FSM+JD [13] (2022)	0.77 *(.06)*	0.23 *(.12)*	0.63 *(.03)*	10.00	4.6		✓	✓	
Causal TCN [13] (2022)	0.80 *(.15)*	0.29 *(.22)*	0.68 *(.24)*	19.00	28.0	✓		✓	
DDNet [31] (2019)	0.88 *(.06)*	0.16 *(.18)*	0.78 *(.14)*	8.00	2.2		✓	✓	
FG + SDN	0.86 *(.06)*	0.17 *(.06)*	0.75 *(.11)*	8.00	4.1		✓		✓
FG + SDN + GC	0.90 *(.11)*	0.10 *(.11)*	0.83 *(.13)*	8.00	4.0		✓		✓
OO-dMVMT	**0.92** ***(.06)***	**0.09** ***(.09)***	**0.85** ***(.11)***	8.00	4.1		✓		✓

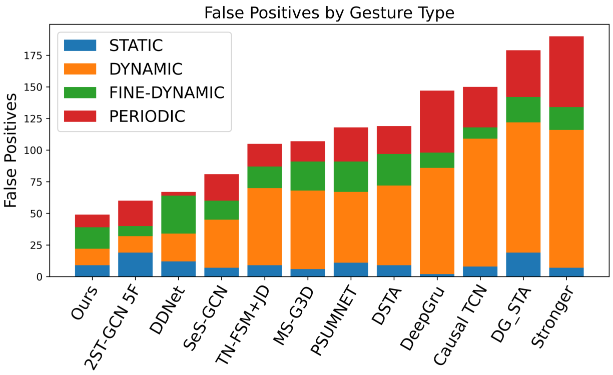

Figure 5. Stacked False Positives for SHREC'22 per model. The gestures are divided into 4 types (colors). The FPs for each type are then summed to show the total value. Lower is better.

see that the number of false positive gestures considered as dynamic gestures diminishes drastically. Our approach has a balanced performance on all the gesture macro categories.

4.3. SHREC'19 results

Results of our tests on the SHREC'19 benchmark [5] are shown in Tab. 2. Some of the metrics are missing due to the unavailability in the original evaluations. The metrics differ slightly from the ones used in SHREC'22 [13] due to the differences between SHREC'19 benchmark to its '22 counterpart, as discussed in Sec. 4.1. For these tests, we chose a larger window size, $W = 40$. This is motivated by the longer average length of the gestures ($\sim$ 45 frames) and by the fact that no short gestures are included in the dictionary. The results are consistent with the ones presented in Sec. 4.2 and show our model achieving state-of-the-art performances on almost all of the metrics, except for the inference time.

Table 2. The metrics are the detection rate , false positives scores , and inference time. Parameters with (*) comes from [5] and may not be accurate. Dash (–) values were not reported in the original benchmark. Standard deviation is reported between brackets.

Method	**DR** ↑	**FP** ↓	**time(ms)** ↓
PSUMNET [29]	0.64 *(.21)*	0.22 *(.20)*	25.0
MS-G3D [22]	0.69 *(.24)*	0.25 *(.22)*	30.3
SeS-GCN [25]	0.75 *(.20)*	0.12 *(.13)*	2.0
SW 3-cent [6]	0.76 *(–)*	0.19 *(–)*	3.0*
DSTA [27]	0.81 *(.11)*	0.08 *(.07)*	8.8
DG-STA [8]	0.81 *(.11)*	0.07 *(.05)*	4.2
DDNet [31]	0.82 *(.13)*	0.10 *(.09)*	2.2
uDeepGRU [5]	0.85 *(–)*	0.10 *(–)*	3.0*
OO-dMVMT	**0.88** ***(.04)***	**0.05** ***(.04)***	5.8

4.4. Ablation studies

In this section, we show the impacts of our Multi-Task Learning On-Off paradigm, as well as discuss the impact of a correct training procedure for the regression heads in a Missing View scenario.

Table 3. Classification results on SHREC'22 with different Multi-task heads. Standard deviation is reported between brackets.

Method	DR ↑	FP ↓	JI ↑
FG	0.88 *(.06)*	0.16 *(.18)*	0.78 *(.14)*
FG + GS/GE	0.52 *(.35)*	0.48 *(.31)*	0.38 *(.32)*
FG + SDN	0.86 *(.06)*	0.17 *(.06)*	0.75 *(.11)*
FG + SDN + GC	0.90 *(.11)*	0.10 *(.11)*	0.83 *(.13)*
OO-dMVMT	**0.92** ***(.06)***	**0.09** ***(.09)***	**0.85** ***(.11)***

4.4.1 Task head removal

In this study, we explore the performance of OO-dMVMT with some tasks removed. Results are in Tab. 3, where FG stands for fine-grained classification head only, FG + GS/GE adds the regression heads, and FG + SDN adds the coarse classification head. The FG model shows better performances than most methods of Tab. 1. In the table we can see a pattern suggesting that, by introducing the multi-view approach, the performances tend to increase. The SDN module slightly decreases the performances but ameliorates the standard deviations. Surprisingly, the sole addition of the regression heads to the FG head drastically reduces the performances. This is probably due to the low task relatedness (start/end gestures are independent of the specific gesture classes), creating competing gradients during the model optimization. However, adding the SDN head allowed for reaching the optimal balance between classification and regression, with the latter improving the overall performance. It is worth noting that a grid search for the weights of the single tasks has been carried out, showing that uniform weighting was the best setup, and that each different task had a role in defining the optimal solution. Therefore, our joint OO-dMVMT appears to be the right task ensemble. Additionally, we removed the regression heads in favor of a binary classifier (GC) trained to classify if a window contains the start or end of a gesture. As opposed to the always-active classification performed by GC, the On-Off regression procedure produces better performance and more stable results.

4.4.2 The On-Off regression head

Since the regression heads are the ones switched on and off by OO-dMVMT, we further investigate them. In particular, we trained with different On-Off policies for the regression heads. The Index Error simulates inaccurate gesture segmentation by randomly changing the gesture start/end index in window ξ_t containing them. The Window Error is more dramatic since it randomly switches on/off the heads with a probability $p = 0.5$. It induces two kinds of errors since a wrongly activated window also implies a wrongly indexed gesture start. In Tab. 4 the results confirm our expectations: both errors degrade the performance, with the Window Error having a worse impact compared to the Index Error.

Table 4. Classification results on SHREC'22 with different Regression Heads training procedures. Standard deviation is reported between brackets.

Method	DR ↑	FP ↓	JI ↑
Window Error	0.76 *(.33)*	0.24 *(.55)*	0.65 *(.31)*
Index Error	0.83 *(.14)*	0.20 *(.21)*	0.72 *(.19)*
OO-dMVMT	**0.92** ***(.06)***	**0.09** ***(.09)***	**0.85** ***(.11)***

5. Limitations

Despite the results being SotA, the OO-dMVMT framework presents a few limitations. Even though we achieve the best false positive performances as highlighted in Tab. 1 we note that, for the fine-dynamic gestures, the false positives are higher than some of the comparative approaches. More research is needed to improve it with additional views or determine if hand acquisition noise is the issue. Although our method has a longer execution time compared to other methods, it is still fast enough to be suitable for real-time applications. Finally, the delay of our approach is heavily dependent on the value of W as explained in Sec. 3.4.

6. Conclusions

Our OO-dMVMT paradigm for facing real-time classification and segmentation of complex hand gestures seems to be ideal: the geometric pose of a hand and its movement are views that are naturally complementary; at the same time, the concurrent processing of multiple tasks is intuitive, mimicking how humans decode a gesture. For example, shaking hands requires that one understands both the start of the gesture from the other partner (to initiate a proper reaction) and the nature of the gesture. The SotA results on all the available benchmarks in the literature further promote our proposal. Our study, in addition, may help in the design of novel hand gestures: Fig. 3 indicates that, regardless of the specific approach, actions like "grab" give a local minimum in terms of detection rate. Future work will be devoted to forecasting the gesture, in order to delete the small delay (8 frames) that we are requiring here.

Acknowledgment

This research is under the National Recovery and Resilience Plan (NRRP), Mission 4, Component 2 Investment 1.4, funded by the European Union – NextGeneration EU, project "Interconnected Nord-Est Innovation Ecosystem".

References

[1] Danilo Avola, Marco Bernardi, Luigi Cinque, Gian Luca Foresti, and Cristiano Massaroni. Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures. *IEEE Transactions on Multimedia*, 21(1):234–245, 2018. 1, 3

[2] Inci M Baytas, Ming Yan, Anil K Jain, and Jiayu Zhou. Asynchronous multi-task learning. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 11–20. IEEE, 2016. 4

[3] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2272–2281, 2019. 1

[4] Ariel Caputo, Andrea Giachetti, Simone Soso, Deborah Pintani, Andrea D'Eusanio, Stefano Pini, Guido Borghi, Alessandro Simoni, Roberto Vezzani, Rita Cucchiara, et al. Shrec 2021: Skeleton-based hand gesture recognition in the wild. *Computers & Graphics*, 99:201–211, 2021. 3, 4

[5] Fabio Marco Caputo, S Burato, G Pavan, Théo Voillemin, Hazem Wannous, Jean-Philippe Vandeborre, Mehran Maghoumi, EM Taranta, A Razmjoo, JJ LaViola Jr, et al. Shrec 2019 track: online gesture recognition. In *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019. 2, 3, 5, 7

[6] Fabio M Caputo, Pietro Prebianca, Alessandro Carcangiu, Lucio D Spano, and Andrea Giachetti. Comparing 3d trajectories for simple mid-air gesture recognition. *Computers & Graphics*, 73:17–25, 2018. 7

[7] Xinghao Chen, Guijin Wang, Hengkai Guo, Cairong Zhang, Hang Wang, and Li Zhang. Mfa-net: Motion feature augmented network for dynamic hand gesture recognition from skeletal data. *Sensors*, 19(2):239, 2019. 3

[8] Yuxiao Chen, Long Zhao, Xi Peng, Jianbo Yuan, and Dimitris N Metaxas. Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. *arXiv preprint arXiv:1907.08871*, 2019. 1, 2, 5, 7

[9] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. Skeleton-based dynamic hand gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2016. 2

[10] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, Joris Guerry, Bertrand Le Saux, and David Filliat. Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset. In *3DOR-10th Eurographics Workshop on 3D Object Retrieval*, pages 1–6, 2017. 2

[11] Guillaume Devineau, Fabien Moutarde, Wang Xi, and Jie Yang. Deep learning for hand gesture recognition on skeletal data. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 106–113. IEEE, 2018. 1, 2

[12] Marco Emporio, Ariel Caputo, and Andrea Giachetti. STRONGER: Simple TRajectory-based ONline GEsture Recognizer. In Patrizio Frosini, Daniela Giorgi, Simone Melzi, and Emanuele Rodolà, editors, *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*. The Eurographics Association, 2021. 3

[13] Marco Emporio, Ariel Caputo, Andrea Giachetti, Marco Cristani, Guido Borghi, Andrea D'Eusanio, Minh-Quan Le, Hai-Dang Nguyen, Minh-Triet Tran, Felix Ambellan, et al. Shrec 2022 track on online detection of heterogeneous gestures. *Computers & Graphics*, 107:241–251, 2022. 2, 3, 4, 5, 6, 7

[14] Sergio Escalera, Vassilis Athitsos, and Isabelle Guyon. Challenges in multi-modal gesture recognition. *Gesture recognition*, pages 1–60, 2017. 3

[15] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics (ToG)*, 39(4):87–1, 2020. 6

[16] Shangchen Han, Po-chen Wu, Yubo Zhang, Beibei Liu, Linguang Zhang, Zheng Wang, Weiguang Si, Peizhao Zhang, Yujun Cai, Tomas Hodan, et al. Umetrack: Unified multi-view end-to-end hand tracking for vr. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 6

[17] Jingrui He and Rick Lawrence. A graph-based framework for multi-task multi-view learning. In *ICML*, 2011. 2

[18] Jingxuan Hou, Guijin Wang, Xinghao Chen, Jing-Hao Xue, Rui Zhu, and Huazhong Yang. Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition. In *European Conference on Computer Vision*, pages 273–286. Springer, 2018. 1, 2

[19] Chuankun Li, Shuai Li, Yanbo Gao, Xiang Zhang, and Wanqing Li. A two-stream neural network for pose-based hand gesture recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 2021. 1, 2

[20] Shaochen Li, Zhenyu Liu, Guifang Duan, and Jianrong Tan. Mvhanet: Multi-view hierarchical aggregation network for skeleton-based hand gesture recognition. *Signal, Image and Video Processing*, pages 1–9, 2023. 3

[21] Yong Li, Zihang He, Xiang Ye, Zuguo He, and Kangrong Han. Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition. *EURASIP Journal on Image and Video Processing*, 2019(1):1–7, 2019. 1, 2

[22] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 143–152, 2020. 1, 2, 5, 7

[23] Chun-Ta Lu, Lifang He, Weixiang Shao, Bokai Cao, and Philip S Yu. Multilinear factorization machines for multi-task multi-view learning. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 701–709, 2017. 2

[24] Mehran Maghoumi and Joseph J. LaViola Jr. Deepgru: Deep gesture recognition utility. *CoRR*, abs/1810.12514, 2018. 1, 2, 3, 7

[25] Alessio Sampieri, Guido Maria D'Amely di Melendugno, Andrea Avogaro, Federico Cunico, Francesco Setti, Geri Sk-

enderi, Marco Cristani, and Fabio Galasso. Pose forecasting in industrial human-robot collaboration. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, pages 51–69. Springer, 2022. 1, 5, 7

[26] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR, 2018. 6

[27] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 1, 2, 5, 7

[28] Eugene M Taranta II, Corey R Pittman, Mehran Maghoumi, Mykola Maslych, Yasmine M Moolenaar, and Joseph J Laviola Jr. Machete: Easy, efficient, and precise continuous custom gesture segmentation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 28(1):1–46, 2021. 3

[29] Neel Trivedi and Ravi Kiran Sarvadevabhatla. Psumnet: Unified modality part streams are all you need for efficient pose-based action recognition. *arXiv preprint arXiv:2208.05775*, 2022. 1, 3, 5, 7

[30] Yi-Feng Wu, De-Chuan Zhan, and Yuan Jiang. Dmtmv: a unified learning framework for deep multi-task multi-view learning. In *2018 IEEE international conference on big knowledge (ICBK)*, pages 49–56. IEEE, 2018. 2

[31] Fan Yang, Yang Wu, Sakriani Sakti, and Satoshi Nakamura. Make skeleton-based action recognition model smaller, faster and better. In *Proceedings of the ACM Multimedia Asia*, pages 1–6. 2019. 1, 3, 4, 5, 7

[32] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021. 2

[33] Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun. Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38:43–54, 2017. 2