OMG-Bench: A New Challenging Benchmark for Skeleton-based Online Micro Hand Gesture Recognition – Supplementary Material

Haochen Chang[1], Pengfei Ren[2]*, Buyuan Zhang[3], Da Li[4], Tianhao Han[3], Haoyang Zhang[5,6], Liang Xie[5,6], Hongbo Chen[1], Erwei Yin[5,6]*
[1]School of Systems Science and Engineering, Sun Yat-sen University
[2]Beijing University of Posts and Telecommunications [3]Shanghai Jiao Tong University
[4]Nankai University [5]Defense Innovation Institute, Academy of Military Sciences
[6]Tianjin Artificial Intelligence Innovation Center
changhch5@mail2.sysu.edu.cn, rpf@bupt.edu.cn, yinerwei1985@gmail.com

In the supplementary material, we provide:

- more dataset details in Sec.A,
- more implementation details in Sec.B,
- more experiments in Sec.C.
- a video demo in Sec.D.

A. More Dataset Details

A.1. Class Definition

We define 40 common single-hand micro gesture classes, as illustrated in Table 1. These gestures are all realized through interactions between the thumb and other fingers. The class differences are characterized by three dimensions: **(1) Interacting fingers**, i.e., the thumb interacting with the index, middle, ring, or little finger. **(2) Interaction types**, including typical actions such as single tap, double tap, slide, pinch, and release. **(3) Interaction locations**, referring to the specific contact areas between the thumb and finger—for example, taps occur at the TIP or MCP joint, while sliding covers a larger finger region. Figure 1 illustrates the interaction locations and types of all micro gestures. Due to the compact spatial arrangement of hand joints and the small temporal differences between different gesture types (e.g., single tap and double tap), varying degrees of confusion arise in both spatial and temporal dimensions, which increase the challenges of online micro gesture recognition.

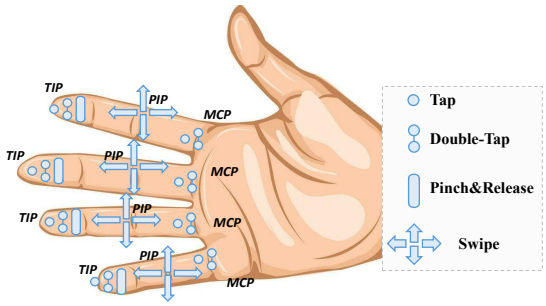

Figure 1. Types and interaction locations of all defined micro gestures. TIP, PIP, and MCP are anatomical terms for finger parts, referring to the finger tip, proximal interphalangeal joint, and metacarpophalangeal joint respectively.

A.2. Data Collection and Annotation

Interactive Tasks for Data Collection. All subjects collected micro gesture sequences by performing randomly generated interactive tasks. Before the data collection began, each subject was informed of the data collection requirements and provided written informed consent for academic use. To ensure data quality, all subjects underwent standardized training before collection to familiarize themselves with the gesture standards. Sequences with non-standard motions or subject anomalies were immediately flagged and repeated.

The interactive task was designed as a gesture-triggered "grid-walking" game, as illustrated in Figure 2. Subjects were required to sequentially move a virtual block along a predefined grid sequence, where each grid cell required the completion of a specific gesture to proceed to the next. A gesture sequence collection was completed once all grid cells were traversed. Each execution path on the grid was randomly generated, and the generated gesture classes varied accordingly. Typically, a sequence contained 8 to 16 micro gestures to ensure diversity within the gesture sequences. To align with human intuitive interaction, the interactive task tends to generate gestures that are consistent with the actual movement of the block during micro gesture generation. For example, a right-swipe gesture controlled

*Corresponding Author.

Table 1. All gesture classes and their corresponding descriptions defined in the OMG-Bench dataset.

Class id	Finger	Location	Type	Class description
1	Index	TIP	Tap	Thumb taps TIP of the index finger
2	Index	MCP	Tap	Thumb taps MCP of the index finger
3	Middle	TIP	Tap	Thumb taps TIP of the middle finger
4	Middle	MCP	Tap	Thumb taps MCP of the middle finger
5	Ring	TIP	Tap	Thumb taps TIP of the ring finger
6	Ring	MCP	Tap	Thumb taps MCP of the ring finger
7	Little	TIP	Tap	Thumb taps TIP of the little finger
8	Little	MCP	Tap	Thumb taps MCP of the little finger
9	Index	TIP	Double-Tap	Thumb double-taps TIP of the index finger
10	Index	MCP	Double-Tap	Thumb double-taps MCP of the index finger
11	Middle	TIP	Double-Tap	Thumb double-taps TIP of the middle finger
12	Middle	MCP	Double-Tap	Thumb double-taps MCP of the middle finger
13	Ring	TIP	Double-Tap	Thumb double-taps TIP of the ring finger
14	Ring	MCP	Double-Tap	Thumb double-taps MCP of the ring finger
15	Little	TIP	Double-Tap	Thumb double-taps TIP of the little finger
16	Little	MCP	Double-Tap	Thumb double-taps MCP of the little finger
17	Index	TIP→MCP	Swipe	Thumb swipes right along the index finger
18	Index	MCP→TIP	Swipe	Thumb swipes left along the index finger
19	Middle	TIP→MCP	Swipe	Thumb swipes right along the middle finger
20	Middle	MCP→TIP	Swipe	Thumb swipes left along the middle finger
21	Ring	TIP→MCP	Swipe	Thumb swipes right along the ring finger
22	Ring	MCP→TIP	Swipe	Thumb swipes left along the ring finger
23	Little	TIP→MCP	Swipe	Thumb swipes right along the little finger
24	Little	MCP→TIP	Swipe	Thumb swipes left along the little finger
25	Index	PIP	Swipe	Thumb swipes up/forward along the index finger
26	Index	PIP	Swipe	Thumb swipes down/backward along the index finger
27	Middle	PIP	Swipe	Thumb swipes up/forward along the middle finger
28	Middle	PIP	Swipe	Thumb swipes down/backward along the middle finger
29	Ring	PIP	Swipe	Thumb swipes up/forward along the ring finger
30	Ring	PIP	Swipe	Thumb swipes down/backward along the ring finger
31	Little	PIP	Swipe	Thumb swipes up/forward along the little finger
32	Little	PIP	Swipe	Thumb swipes down/backward along the little finger
33	Index	TIP	Pinch&Release	Thumb and index finger pinch
34	Index	TIP	Pinch&Release	Thumb and index finger release
35	Middle	TIP	Pinch&Release	Thumb and middle finger pinch
36	Middle	TIP	Pinch&Release	Thumb and middle finger release
37	Ring	TIP	Pinch&Release	Thumb and ring finger pinch
38	Ring	TIP	Pinch&Release	Thumb and ring finger release
39	Little	TIP	Pinch&Release	Thumb and little finger pinch
40	Little	TIP	Pinch&Release	Thumb and little finger release

the block to move right, while single and double taps controlled turning. As a result, the dataset contains some consecutively performed gestures of the same class, which better reflects realistic VR/AR interaction scenarios.

To enable subjects to trigger interactive tasks without an online micro gesture recognition model, we designed a threshold-based heuristic micro gesture recognition algorithm. This heuristic algorithm identifies gestures by measuring the distances between the thumb and specific joints of other fingers. Our primary objectives were (1) to provide subjects with real-time interactive feedback to ensure natural gesture execution, and (2) to determine the start and end of gestures via the heuristic algorithm, enabling automated gesture frame labeling. Therefore, the classification accuracy of the heuristic algorithm was not critical; rather, its accuracy in detecting gesture occurrences was essential.

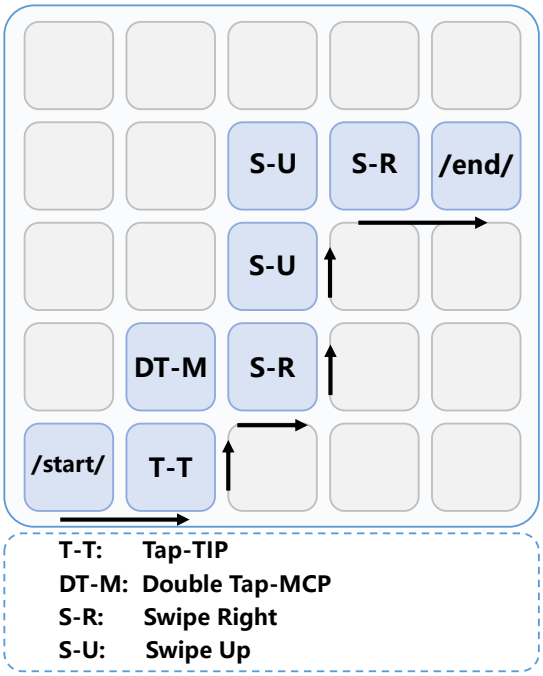

Figure 2. Illustration of the interactive task.

For example, since the system knows that the target gesture is a thumb sliding along the index finger, it can determine the start and end frames by measuring the distance between the thumb tip and either the index fingertip or the base of the index finger, without considering the other fingers. This greatly reduces the difficulty of heuristic recognition, thereby maximizing the chance that each gesture triggers the virtual module to move to the next grid cell. Finally, we retain only the gesture boundaries predicted by the heuristic algorithm, while the gesture labels are directly assigned from the predefined sequence.

Human Annotation. Five professional volunteers familiar with gesture interaction and motion annotation were recruited to inspect and correct any erroneous automatic labels, ensuring label accuracy. Specifically, experts review skeleton-overlaid RGB frames to rectify class labels and boundary inaccuracies. If the gestures performed by subjects differed from predefined sequence, predefined labels were modified to match. If initial boundaries differ significantly from the actual motion, they are manually re-annotated. Compared to traditional labor-intensive manual annotation methods, our approach significantly improved accuracy and avoided uncertainty caused by human bias.

Inter-Annotator Consistency. We invited five experts to jointly annotate 5% of the data and computed the pairwise boundary IoU and boundary variance among them. The average boundary IoU was 0.81, and the average boundary variance was 2.13 frames, indicating high annotation consistency. Furthermore, after retraining with random boundary noise applied to the ground truth (±5 frames, p = 0.5), HMATr achieved a DR of 89.1% (a drop of 0.1%), while OO-dMVMT achieved a DR of 78.7% (a drop of 0.4%). These results demonstrate that annotation noise within 5 frames has a negligible impact on training.

Multi-view Hand Pose Automatic Annotation Algorithm. OMG-Bench follows the standard 21-joint skeleton annotation scheme [14], which includes the fingertip and three phalangeal joint centers for each finger, along with the wrist joint. However, annotating large-scale hand pose datasets is both time-consuming and labor-intensive. Therefore, we employed a multi-view self-supervised hand pose estimation method to achieve automatic hand skeletal acquisition. A self-supervised keypoint detector is trained on depth images using the Dual-Branch Self-Boosting Framework (DSF) [11], and applied to unlabeled multi-view sequences following the Cross-View Fusion Network strategy [12]. This automatic stage yields accurate and consistent 3D joint coordinates and MANO parameters under most conditions. We tested this method on the manually annotated held-out evaluation set, and the error is 2.78 mm.

A.3. Dataset Attribute Metrics

In Table 2, we use four metrics to measure the differences between OMG-Bench and SHREC'21 [1] and SHREC'22 [6].

- **Same-Class Continuous Gesture Percentage (SC-CGP)**: The proportion of continuously performed gestures belonging to the same class relative to the total number of samples. OMG-Bench contains more consecutive gestures of the same class, better reflecting real-world VR/AR interaction scenarios. For example, when users browse web pages or select applications from VR virtual menus, they often perform consecutive downward swipe gestures to turn pages. At the same time, these consecutive same-class gestures are prone to being recognized as a single gesture, increasing boundary ambiguity in online gesture recognition.
- **Mean Gesture Interval (MGI)**: The average time interval between consecutive gesture samples. The gesture intervals in OMG-Bench are relatively shorter, better reflecting the natural gesture interaction frequency of humans in VR/AR environments, but they also further increase boundary ambiguity in online gesture recognition.
- **Mean Gesture Duration (MGD)**: The average duration of all gesture samples. This metric indicates that most gestures in OMG-Bench have significantly shorter durations compared to the other two datasets. Such a short response time characteristic of micro gestures increases the challenge of online recognition.
- **Normalized Mean Joint Displacement (NMJD)**: The average joint displacement over all sequences and joints, normalized by the distance between the wrist joint and the

Table 2. Statistical Comparison of OMG-Bench with SHREC'21/22.

	SCCGP↑	MGI↓	MGD↓	NMJD↓
SHREC'21 [1]	0.29%	12.48s	2.60s	158.38
SHREC'22 [6]	0.09%	2.86s	1.19s	128.73
OMG-Bench	**27.60%**	**0.22s**	**0.57s**	**8.95**

root of the middle finger. This suggests that, compared to the other two datasets, OMG-Bench exhibits more subtle motion characteristics, making it challenging to precisely determine the position and class of micro gestures from fine-grained movements.

B. More Implementation Details

B.1. Training Details

We implement our method with PyTorch framework and perform all experiments on one RTX4090D GPU. We train our models using Adam with a weight decay of 0.0004. The batch size is set to 64 and the base learning rate is set to 0.001. We adopted a cosine learning rate scheduling strategy with a 5-epoch warm-up phase. The loss weights were set through grid search as follows: $\lambda_{cls} = 2$, $\lambda_{pos} = 5$, $\lambda_{q-CTC} = 0.2$. The number of learnable queries is set to 10, the length of the frame-level memory bank is set to 16, and the length of the window-level memory bank is set to 3. The Frame Memory Interaction, Query Memory Interaction, and Position-aware Interaction modules were each configured with 1 layer. The number of attention heads in both the self-attention and cross-attention modules is set to 4. The classification confidence threshold during online inference is set to 0.7. The feature extraction backbone comprises 3 ST-GCN layers [13] with channel configurations 3→64, 64→128, and 128→256.

Since the length of each gesture sequence varies, we divided each sequence into shorter segments of 128 frames for temporal training. The sliding step for segmentation was set to 64 frames to ensure effective utilization of all data. Within each short segment, we applied a non-overlapping sliding window scheme with a window size of 16 frames and a stride of 16 frames to train the proposed HMATr. As a result, each short segment contained 8 continuous windows.

For other baselines, we follow the training paradigm commonly adopted in typical skeleton-based human action recognition methods [2–4, 8] as well as previous online gesture recognition approaches [5] for all compared methods. Specifically, we first perform offline training on the pre-segmented dataset, and then adopt a sliding-window strategy during online inference, with a window size of 16, a stride of 1, and a Logits score threshold of 0.8.

B.2. Query-based CTC loss

We employed an auxiliary loss $\mathcal{L}_{q-CTC}$ based on Connectionist Temporal Classification (CTC) [7] alongside the Hungarian matching loss. Specifically, we supervised the classification results of matched queries within each 128-frame short segment using the standard CTC loss. This leverages CTC's ability to predict blank tokens to suppress outputs from irrelevant queries, thereby further enhancing the queries' capability to perceive gesture boundaries. Related ablation studies are presented in Sec.C.1.

B.3. Benchmark Evaluation Metrics

We employed six metrics to evaluate the performance of the model.

- **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 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.
- **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.
- **Normalized Levenshtein Distance (NLD)**: Defined as the average normalized edit distance between the predicted gesture sequence and the ground truth sequence. It measures the minimum number of insertions, deletions, and substitutions required to transform the predicted sequence into the ground truth, normalized by the length of the ground truth. NLD evaluates the overall sequence-level similarity and penalizes recognition errors including missed, spurious, and misclassified gestures. It can be expressed as:

$$\text{NLD} = 1 - \frac{\text{levenshtein}(y_{\text{predict}}, y_{\text{true}})}{\text{length}(y_{\text{true}})}, \tag{1}$$

where y_{predict} and y_{true} are the predicted and true list of labels of the gestures respectively.

- **Inference Time**: We perform model inference on an RTX 4090D GPU and measure the inference time (in milliseconds) with a batch size of 1. This metric serves as the most direct measure of the efficiency of online gesture recognition models.
- **Average Delay**: Average Delay is defined as the difference, measured in frames, between the actual gesture end

Table 3. Ablation study of different memory bank lengths.

Frame	Window	DR↑	FP↓	JI↑	NLD↑
1	3	85.9%	0.28	0.63	0.70
8	3	87.2%	0.24	0.68	0.75
16	**3**	**89.2%**	**0.22**	**0.71**	0.77
24	3	88.9%	0.22	0.66	**0.79**
16	1	87.6%	0.23	0.66	0.74
16	2	87.5%	0.22	0.68	0.75
16	4	88.2%	0.25	0.69	0.75

Table 4. Ablation study of the number of queries.

Number of Queries	DR↑	FP↓	JI↑	NLD↑
3	85.6%	**0.21**	0.65	0.76
5	86.2%	0.23	0.67	0.75
10	**89.2%**	0.22	**0.71**	**0.77**
20	88.5%	0.26	0.66	0.74

Table 5. Ablation study of Query-CTC loss.

Loss	DR↑	FP↓	JI↑	NLD↑
w/o $\mathcal{L}_{q-CTC}$	88.8%	0.25	0.69	0.73
Ours	**89.2%**	**0.22**	**0.71**	**0.77**

frame and the last frame reported by the algorithm for predicting the gesture start frame.

C. More Experiments

C.1. Ablation Studies

Effect of Memory Bank Length. We investigate the impact of memory bank length on online micro gesture recognition by fixing the length of one memory bank while varying the length of the other. Results in Table 3 show that the best performance is achieved when the frame-level memory bank length is 16 and the window-level memory bank length is 3. We found that neither a memory bank that is too short nor one that is too long yields the best performance. This is because when the memory bank is too short, the model obtains limited historical temporal information from it. Conversely, when the memory bank is too long, the historical information may contain multiple completed independent gestures, which can interfere with the current gesture prediction.

Effect of the Number of Queries. We conduct ablation studies to investigate the effect of the number of queries on online micro gesture recognition. Experiments are performed with the number of queries set to 3, 5, 10, and 20. Results in Table 4 show that the overall performance is best when the number of queries is 10. When the number of queries is too small, their diversity in representing positional and semantic features is limited, making it difficult to capture the complexity and variability of micro gestures. Meanwhile, their positional awareness is relatively sparse, failing to represent multiple possible locations. Conversely, when the number of queries is too large, irrelevant queries may be activated, resulting in redundant gesture detections.

Effectiveness of Query-CTC Loss. We validate the effectiveness of the Query-CTC loss through ablation experiments. Table 5 presents the results of experiments without the Query-CTC loss. We observe that removing the Query-CTC loss significantly affects the normalized Levenshtein distance (NLD), while other metrics show minor changes. This demonstrates that the Query-CTC loss primarily optimizes the network learning from the perspective of the overall sequence. The Query-CTC loss can suppress the activation of irrelevant queries to some extent, preventing redundant gesture outputs. Additionally, it injects global semantic information into the position-aware queries at the sequence level, enhancing the feature representation capability of the queries.

Table 6. Ablation study of global memory embedding strategy.

Strategy	DR↑	FP↓	JI↑	NLD↑
Cross-Att.	89.0%	**0.20**	0.70	0.76
Zero init.	88.5%	0.24	0.68	0.72
Memory init. (Ours)	**89.2%**	0.22	**0.71**	**0.77**

Effect of Global Memory Embedding Strategy. To investigate how to effectively exploit the global historical information contained in the window-level memory, we explore three global memory embedding strategies for the queries: (1) Zero initialization: no global history is embedded; position-aware queries are directly zero-initialized. (2) Cross-attention: position-aware queries attend to the window-level memory features via cross-attention to inject global historical information. (3) Memory initialization: position-aware queries are initialized with the mean-pooled feature of the window-level memory bank (Section 4.3 in the main text). As shown in Table 6, our memory initialization achieves the best performance for embedding historical information. Moreover, compared to cross-attention, memory initialization introduces no additional learnable parameters and is therefore more efficient.

C.2. More Quantitative Experiments

Effect of Skeleton Accuracy in the Dataset. To evaluate the effect of skeletal data accuracy in the dataset on the performance of HMATr, we conducted inference-stage experiments on test sets with varying levels of skeletal data precision. Specifically, we first generate hand skeleton sequences for the entire test set using different hand pose esti-

Table 7. Results of inference using HMATr trained on the original training set on test sets derived from different Hand Pose Estimation (HPE) algorithms.

HPE Method	Modality	DR↑	FP↓	JI↑	NLD↑
WiLoR [10]	RGB	73.6%	0.15	0.64	0.68
HaMeR [9]	RGB	77.3%	0.14	0.67	0.73
MMVI-single [12]	Depth	83.7%	0.20	0.73	0.77

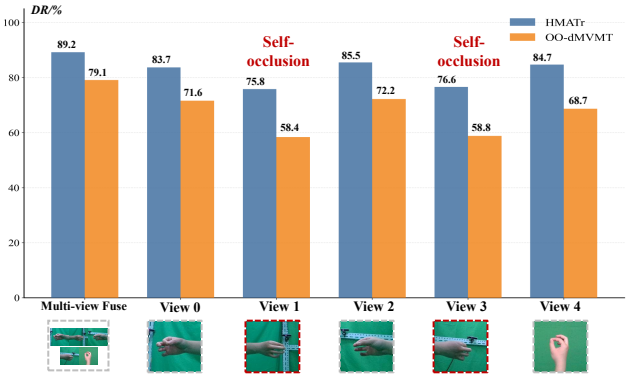

Figure 3. Inference results of HMATr and OO-dMVMT trained on OMG-Bench under single-view settings. The skeleton data for each view are obtained using a single-view hand pose estimation algorithm to simulate the inaccuracies of skeleton estimation in real-world applications.

mation algorithms (e.g., WiLoR [10], HaMeR [9], MMVI-single [12]), relying solely on monocular RGB-D data from camera 0. We then directly perform inference on these skeleton sequences — produced by various pose estimation algorithms — using HMATr trained on the OMG-Bench training set. In other words, the training data consist of skeletons obtained via multi-view fusion from the original OMG-Bench dataset, whereas the testing data consist of skeletons estimated from single-view inputs by different algorithms.

As shown in Table 7, the accuracy drop of HMATr varies across test sets with different skeletal data qualities, which is primarily constrained by the quality of skeletal estimations in the test set. MMVI-single [12], a single-view variant of our multi-view fusion algorithm for skeletal data generation, produces skeletal estimations of relatively higher quality compared to the RGB-based methods WiLoR [10] and HaMeR [9], thereby achieving comparatively higher testing accuracy. Moreover, the limited accuracy degradation observed in Table 7 also reflects, to some extent, the generalization capability of HMATr, indicating its applicability to skeletal data estimated by different algorithms.

Single-view generalization. During training, we apply skeleton rotation and translation normalization to improve

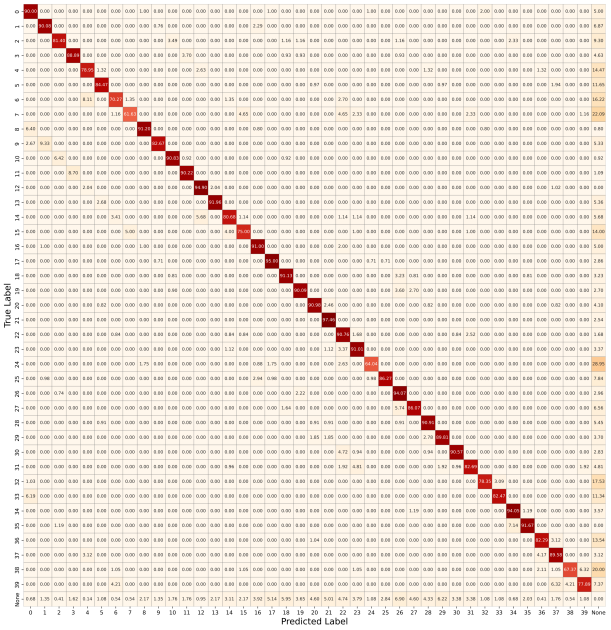

Figure 4. The confusion matrix of HMATr on OMG-Bench.

the robustness of our method to large-amplitude motions and arbitrary viewpoints. To better reflect real-world conditions, we perform inference using skeletons estimated from each individual view. Figure 3 presents the test results under different viewpoints. We observe that even in extreme views such as View1/3, where severe self-occlusion occurs, HMATr still maintains a satisfactory gesture detection rate, demonstrating its robustness. Meanwhile, the highest recognition accuracy is achieved under multi-view fusion, highlighting the necessity of acquiring high-quality skeleton data through multiple views.

C.3. Confusion Analysis

Figure 4 illustrates the confusion patterns of our HMATr for online micro-gesture recognition on the OMG-Bench dataset. We identify the top three most confused class pairs as class 9/class 1, class 6/class 4, and class 11/class 3. Analysis of these three challenging pairs reveals two main difficulties: (1) temporal confusion: rapid double-tap gestures of the index and middle fingers are often misclassified as single taps, with error rates of 9.86% and 8.79%, respectively, due to the short temporal intervals in continuous inputs; (2) anatomical confusion: tapping with the little finger is easily confused with tapping with the ring finger (error rate: 9.68%), owing to the strong coupled motion between these two adjacent fingers.

D. Video Demo

In our project page, we provide a video to further illustrate visualization details in the OMG-Bench dataset and

the practical deployment of the proposed HMATr.

The video showcases samples from the 40 micro gesture classes included in OMG-Bench, covering four common interaction types: Swipe, Tap, Double-Tap, and Pinch&Release. The inherent properties of OMG-Bench — rapid dynamics, subtle motion, high inter-class similarity, and frequent same-class continuity — pose new challenges for online micro gesture recognition.

Additionally, we deploy HMATr, trained on the OMG-Bench dataset, onto the Quest Pro headset to evaluate practical micro gesture interaction. We directly leverage the headset's built-in high-precision hand pose estimation to acquire skeletal data in real time. The generalization primarily stems from skeleton rotation and translation normalization applied during training and from HMATr's hierarchical memory mechanism with position-aware queries. Specifically, the hierarchical memory leverages historical context to effectively smooth inter-frame jitter and occlusion-induced noise prevalent in monocular estimation, while the attention-based position-aware queries focus on relative motion patterns rather than absolute coordinates, enabling implicit semantic alignment across heterogeneous skeleton structures and thereby ensuring robust cross-algorithm inference. Moreover, the high-precision skeleton data in OMG-Bench further contributes to this generalization. The video demonstrates that the proposed online micro gesture recognition algorithm supports interaction via multiple micro gestures while ensuring real-time performance and low latency.

References

[1] 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

[2] Haochen Chang, Jing Chen, Yilin Li, Jixiang Chen, and Xiaofeng Zhang. Wavelet-decoupling contrastive enhancement network for fine-grained skeleton-based action recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4060–4064. IEEE, 2024. 4

[3] Haochen Chang, Pengfei Ren, Haoyang Zhang, Liang Xie, Hongbo Chen, and Erwei Yin. Hierarchical-aware orthogonal disentanglement framework for fine-grained skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11252–11261, 2025.

[4] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13359–13368, 2021. 4

[5] Federico Cunico, Federico Girella, Andrea Avogaro, Marco Emporio, Andrea Giachetti, and Marco Cristani. Oo-dmvmt: A deep multi-view multi-task classification framework for real-time 3d hand gesture classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2745–2754, 2023. 4

[6] 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. 3, 4

[7] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006. 4

[8] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoun Lee. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10444–10453, 2023. 4

[9] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9826–9836, 2024. 6

[10] Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12242–12254, 2025. 6

[11] Pengfei Ren, Haifeng Sun, Jiachang Hao, Qi Qi, Jingyu Wang, and Jianxin Liao. A dual-branch self-boosting framework for self-supervised 3d hand pose estimation. *IEEE Transactions on Image Processing*, 31:5052–5066, 2022. 3

[12] Pengfei Ren, Haifeng Sun, Jiachang Hao, Jingyu Wang, Qi Qi, and Jianxin Liao. Mining multi-view information: a strong self-supervised framework for depth-based 3d hand pose and mesh estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20555–20565, 2022. 3, 6

[13] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 4

[14] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3d hand pose tracking and estimation using stereo matching. *arXiv preprint arXiv:1610.07214*, 2016. 3