Towards Seamless Egocentric Hand Action Recognition in Mixed Reality

Anonymous Author 1*
Organization 1

Anonymous Author 2†
Organization 2

Anonymous Author 2‡
Organization 2

Anonymous Author 2§
Organization 2

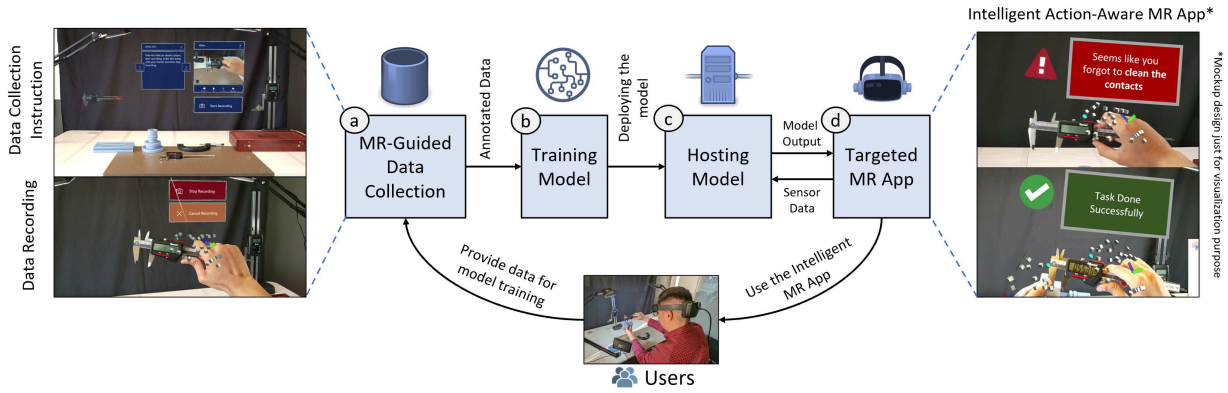


Figure 1: Hand action recognition ecosystem for intelligent mixed reality (MR) applications. (a) MR-guided data collection app with various modes of instruction and automated annotation support for efficient dataset generation. (b) Training collected dataset with our robust and efficient action recognition model. (c) Deploying the trained model either on a headset directly or a remote server for enhanced performance and flexibility. (d) Targeted MR app supported by real-time action understanding capability.

Abstract

Understanding user actions from egocentric videos is crucial in developing intelligent mixed reality (MR) systems. One key aspect is the recognition of hand actions and gestures, which enables user interaction and adaptation of the system to real-world user actions. In this paper, we present a comprehensive pipeline for egocentric hand action recognition in mixed reality applications. Our approach incorporates an MR-guided data collection method that eliminates the need for explicit manual annotation and guidance. We also propose a robust and efficient skeleton-based hand action recognition model specifically designed for real-time MR use cases. To validate our proposed framework and demonstrate its effectiveness, we conducted a case study involving industrial precision inspection tasks. Utilizing our MR-guided data collection system, we efficiently collected hand inspection action data and built a comprehensive dataset. We then trained our proposed model on this dataset, employing a feature refinement strategy. We conducted extensive evaluations, including standard offline analysis and real-time inference in an MR system, to thoroughly test the model. Our experimental results showcase the efficacy of our proposed pipeline and its potential for practical use in various scenarios. The source code and the dataset are publicly available on [anonymous link].

Index Terms: Human-centered computing—Human-computer interaction (HCI)—Interaction paradigms—Mixed / augmented reality; Computing methodologies—Artificial intelligence—Computer vision—Action recognition

*e-mail: abc1@xyz.com
†e-mail: abc2@xyz.com
‡e-mail: abc2@xyz.com
§e-mail: abc2@xyz.com

1 Introduction

Hand action recognition plays a crucial role in enabling an intelligent and adaptive mixed reality system by helping it understand the user's current activity state and dynamically adapt to their needs. By accurately recognizing and analyzing hand gestures, the system can interpret the user's intentions, preferences, and level of expertise. This information allows the system to provide adaptive guidance and intervention in mixed reality applications. For instance, in educational scenarios, hand action recognition can assist in real-time feedback and guidance, correcting the user's actions and providing instructional cues based on their hand movements. In professional scenarios, such as industrial training, hand action recognition can enable the system to detect errors or unsafe practices, intervening with appropriate warnings or instructions to enhance learning outcomes and ensure user safety [8,9,32,37]. Overall, by understanding the user's hand actions, an intelligent mixed reality system can dynamically adjust its responses, optimizing the user's experience, and enhancing the effectiveness of mixed reality applications.

In recent years, various approaches have been proposed for hand action recognition, utilizing different modalities such as RGB-D video feeds and skeleton data. While RGB-D data provide rich visual information, it requires substantial processing costs due to its high-dimensional nature. Conversely, skeleton data provides a high-level, compact representation of hand actions, enabling fast processing and achieving high accuracy in prediction. Additionally, skeleton data can be a suitable choice in scenarios where security and privacy risks associated with sharing RGB data need to be minimized. Using skeleton data, the focus is shifted to the high-level representation of hand actions, while reducing the transmission of sensitive visual information.

In the context of MR, a significant advantage arises from the utilization of hand skeleton data captured by MR headsets. Unlike general systems that require additional processing to obtain hand skeleton data from RGB-D video feeds using 3D pose estimation algorithms [21, 38], MR headsets inherently process and provide skeleton data for basic interaction with the MR user interface. This inherent availability of hand skeleton data in MR headsets minimizes the need for extra processing, facilitating efficient hand action

recognition within the MR environment.

Despite the proliferation of research on skeleton-based action recognition (discussed in Section 2), few studies have focused specifically on the unique requirements of hand action recognition for MR systems. Consequently, there is a gap in the literature regarding efficient data collection methodologies, appropriate recognition models, deployment strategies, and integration of hand action recognition into MR applications.

In this paper, we propose a comprehensive framework for hand action recognition in MR applications, addressing the aforementioned research gap from an ecosystem perspective. Our framework serves as a plug-and-play solution, providing users with the flexibility to customize and scale it for various MR purposes. Key contributions of our work include:

- An efficient MR-guided hand action data collection app that eliminates the need for explicit data annotation or curation, empowering users to collect high-quality data without requiring external supervision.
- A comprehensive hand action recognition dataset, leveraging our proposed data collection app for a representative use case, facilitating benchmarking and further advancements in the field.
- A robust and compact hand action recognition model that incorporates state-of-the-art training and deployment techniques, ensuring accurate and real-time recognition performance.
- Rigorous experiments and ablation studies to evaluate the proposed components, demonstrating the effectiveness and applicability of our framework.

2 Related Works

In the rapidly evolving field of mixed reality, extensive research has been conducted on hand data analysis, hand gesture recognition, and action recognition, paving the way for intuitive and interactive user interfaces and facilitating intelligent user interaction in mixed reality environments. Several notable contributions have emerged in this space. Malik et al. [19] pioneered hand tracking techniques for interactive pattern-based augmented reality (AR), while Xie et al. [34] developed a lightweight and efficient online gesture recognition network tailored for embedded AR devices. Capece et al. [1] focused on devising an easy hand gesture recognition system for collaborative experiences in extended reality, and Khurshid et al. [13] explored hand gesture recognition to improve user interaction in AR settings. Murhij et al. [20] devised a dedicated hand gesture recognition model specifically targeted at AR robotic applications. Furthermore, Wang et al. [30] and Che et al. [2] made significant contributions by designing gesture recognition algorithms to augment interaction in driving systems and mobile devices, respectively. Ryu et al. [25] proposed methods for recognizing and correcting actions in virtual and augmented reality, while Patil et al. [23] focused on human action recognition utilizing skeleton features. Lu et al. [18] developed techniques to automate editing processes in mixed reality videos based on hand-object interactions. Furthermore, the availability of datasets such as FirstPiano (Voillemin et al. [29]) has facilitated research in hand action recognition for AR applications. Furthermore, Wolf et al. [33] utilized gaze tracking to predict and prevent erroneous hand movements during AR-supported manual tasks. Although these contributions have significantly advanced the field, there is still considerable room for further advancements in hand action recognition for intelligent mixed reality applications to address the evolving requirements of this new era.

Hand Action Recognition Datasets. Hand action recognition research has witnessed remarkable progress in recent years, driven by the availability of diverse and well-annotated datasets. These datasets play a crucial role in the advancement of the field by providing researchers with valuable resources to develop and evaluate algorithms. In this regard, several notable datasets have emerged, each catering to specific aspects of hand action recognition. The First-Person Hand Action (FPHA) dataset [12] offers RGB-D videos and 3D hand pose annotations, facilitating first-person perspective hand action recognition. The SHREC'17 track dataset [7] focuses on dynamic hand gestures captured by the Intel RealSense depth camera. The Dynamic Hand Gesture 14/28 (DHG-14/28) dataset [6] provides sequences of hand gestures along with depth images and hand skeleton coordinates. The H2O dataset [14] emphasizes two-handed manipulation of objects for first-person interaction recognition, encompassing RGB-D and 3D hand pose data. For human-robot interaction, the HRI gesture dataset [11] combines RGB, depth, and 3D skeleton data. The Handicraft Dynamic Hand Gesture dataset [17] employs the Leap Motion controller and the hand skeleton data, specifically targeting hand gestures in handicraft activities. Lastly, the AssemblyHands dataset [22] offers a large-scale benchmark with accurate 3D hand pose annotations, focusing on challenging hand-object interactions in egocentric activities. These datasets collectively contribute to the advancement of hand action recognition research and enable researchers to develop robust algorithms across a wide range of applications. However, while these datasets have contributed significantly to the field, there is a need for a systematic data collection mechanism, which is often lacking or not explicitly described in the literature. Moreover, the majority of existing datasets primarily employ static cameras or sensors such as Intel RealSense and Leap Motion. Therefore, there is a lack of exploration regarding the utilization of sensors or direct hand skeleton data from mixed reality headsets. This unexplored area has the potential for significant advancements in hand action recognition research.

Skeleton-based Action Recognition Models. In the realm of skeleton-based action recognition, previous work has approached the task as a sequence classification problem, employing various techniques to extract meaningful features from the skeletal data. One of these approaches focused on the use of recurrent neural networks (RNNs) within auto-encoders to capture high-level features from skeleton sequences [28]. Additionally, some researchers employed convolutional neural networks (CNNs) by converting skeleton data into image-like representations through hand-crafted schemes [9, 37], while others explored temporal convolutional network (TCN) architectures [26, 36]. However, these methods did not explicitly exploit the inherent spatial structure of the skeleton. To address this, researchers turned to graph convolutional networks (GCNs) that take advantage of the natural graph-like structure formed by the joints and bones of the hand or human body. Pioneering work by Yan et al. [35] defined spatial and temporal connections, Zhang et al. [39] introduced a two-stream architecture, and Chen et al. [3] improved upon the GCN design. Recently, Zhou et al. [40] proposed a GCN-based framework that refines features and enhances discriminative representation to alleviate confusion in ambiguous actions for skeleton-based action recognition. In the domain of skeleton-based gesture recognition, Liu et al. [16] proposed a Temporal Decoupling Graph Convolutional Network (TD-GCN) that applies different adjacency matrices for skeletons from different frames to effectively model temporal information. Furthermore, Peng et al. [24] developed an Efficient Graph Convolutional Network (EGCN) for solving the Travelling Salesman Problem (TSP) on 2D Euclidean graphs, leveraging the power of GCNs in constructing efficient graph representations and generating optimal tours in a non-autoregressive manner through highly parallelized beam search. Despite recent advancements, there remains scope for further improvement, particularly in the context of real-time mixed reality applications that require highly accurate, reliable, and fast inference models.

3 METHOD

This section provides a comprehensive discussion of our method, offering detailed insights into our hand action recognition pipeline. Figure 1 presents an overview of the pipeline from an ecosystem perspective.

3.1 Mixed Reality-Guided Data Collection

To efficiently collect hand action recognition data without the need for explicit data annotation and curation effort, we propose a novel MR-guided data collection app. This application leverages text, video, and 3D model/animation instruction modes to guide users in performing various actions. It offers a smooth user experience, allowing individuals to follow instructions and record their actions without requiring external supervision. Figure 2 presents a user-centric flowchart of the app, outlining the step-by-step process, while Figure 1(a) provides a visual depiction of the app, offering glimpses into its interface.

The user begins by opening the app and selecting the first action or choosing a specific action from the available options. They then review the step-by-step instructions provided, which may include textual descriptions, video demonstrations, and animated 3D models. Once familiar with the instructions, the user initiates the recording by tapping the start record button or using a corresponding voice command. During the recording, the user performs the action as instructed. Upon successful completion of the action, the recording can be saved by stopping the recording process. In the event of an undesired mistake or error during the action, the user has the option to cancel the recording by tapping a designated button or issuing a voice command. They can then proceed to the next action following the same process.

Since the app follows a sequential instruction approach, each recorded data segment corresponds to a specific action category. This inherent logging mechanism annotates the action sequence as the user progresses through the app. The comprehensiveness of the instructions, which combine multiple mediums such as text, video, and animation, allows users to independently collect their own data effectively and flawlessly without requiring external supervision.

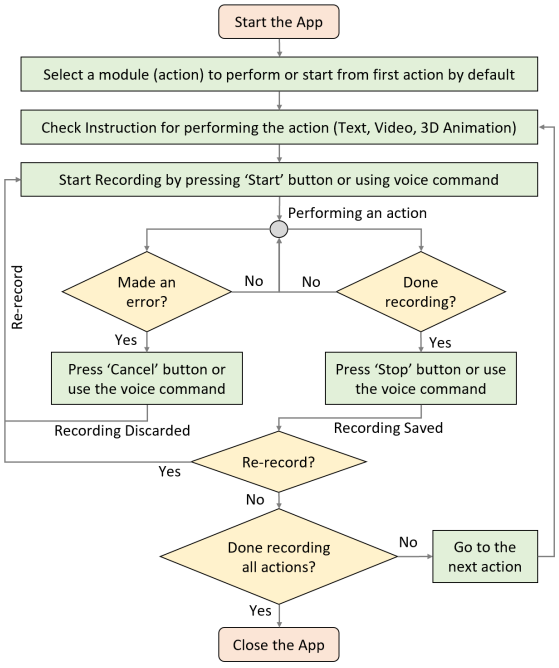


Figure 2: The MR-guided data collection process.

Additionally, the voice interaction feature enhances the user experience by enabling hands-free interaction, allowing users to focus on following instructions and performing actions with their hands.

Our app provides a user-friendly and scalable solution, allowing users to easily customize instructions and generate annotated datasets without manual effort. This flexible approach has great potential to advance hand action recognition through large-scale data collection, offering users the ability to obtain high-quality datasets tailored to their specific needs. By simplifying the data collection process, our app empowers researchers and practitioners in the field to accelerate their work and explore new possibilities.

3.2 Skeleton-Based Hand Action Recognition Model

The proposed hand action recognition model is presented in Figure 3, providing a simplified overview of its architecture. In the sequel, we dive into the specific details and components of the model, offering a more comprehensive understanding of its inner workings.

3.2.1 Input Data

The input data for our model consist of hand skeleton data acquired from a mixed reality headset (e.g., HoloLens 2 in this study). The raw input data is represented as a three-dimensional tensor with dimensions $T \times J \times 3$, where T denotes the number of frames, J represents the number of hand joints, and 3 signifies the joint position values for the three dimensions (x, y, z). To prepare the raw input for our model, we employ a preprocessing technique similar to the approach described in [4]. To ensure a consistent input size for our main model, we transform the data dimensions to $T' \times J \times 3$, where T' denotes a fixed number of frames. We achieve this by sampling T' frames from the original action sequences. In our study, we set T' to be 52 frames. Considering the hand joint information provided by the HoloLens 2 headset, we extract 26 hand joints for our analysis. These joints are illustrated in Figure 4, providing a visual reference for the specific hand joints considered in our study.

3.2.2 Backbone Neural Network Architecture

Our backbone architecture is constructed based on [40] and has been modified to meet the real-time demands of our application. The model consists of 7 core units referred to as Temporal Graph Networks (TGNs). These TGNs are created by combining Temporal Convolutional Networks (TCNs) and Graph Convolution Networks (GCNs). TCNs extract temporal features by utilizing 1D Convolutional Neural Networks (CNNs) along the temporal dimension, while GCNs capture spatial features by leveraging a learnable topological graph defined on the spatial dimension.

Among the 7 fundamental units, one of them is implemented as strided TGNs using strided 1D CNNs. This specific unit facilitates the generation of multi-scale features by reducing the temporal dimension while simultaneously increasing the channel dimension. Following this, a pooling layer is applied to obtain high-level 1D feature vectors. Finally, these features are mapped to a probability distribution over K candidate categories using a fully connected (FC) layer with softmax activation.

It is worth noting that the architecture intentionally maintains a generic structure, allowing for the substitution of the basic unit's implementation with other GCN-based networks [3, 15, 27, 31, 35]. By incorporating this flexibility, our approach encourages the exploration and utilization of various GCN-based architectures within our framework.

3.2.3 Feature Refinement

We adopt an innovative training strategy from [40] and implemented it in the context of hand action recognition to enhance our feature representation. Our goal is to improve the performance of the skeleton-based model, specifically for ambiguous actions that are easily misclassified due to their similarities. To address this, we

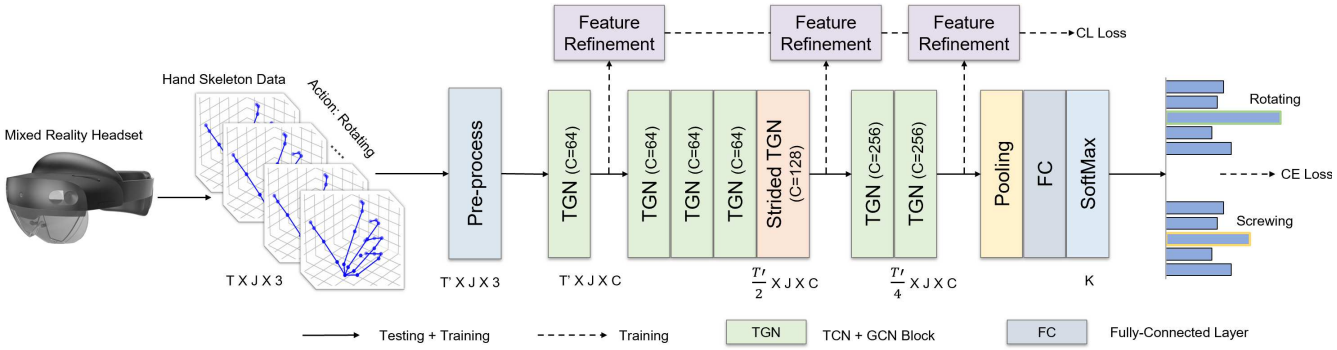


Figure 3: Proposed hand action recognition model.

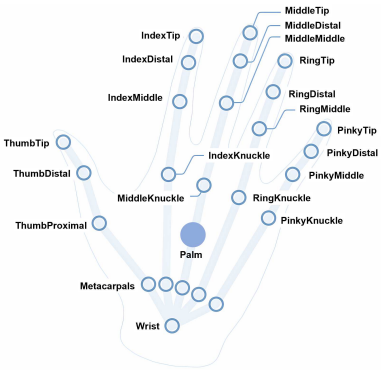
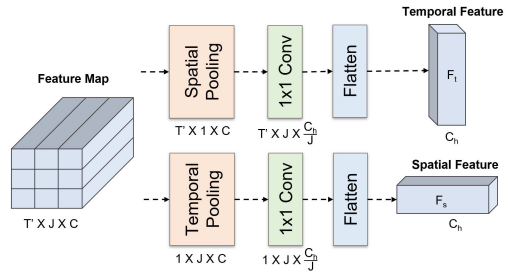
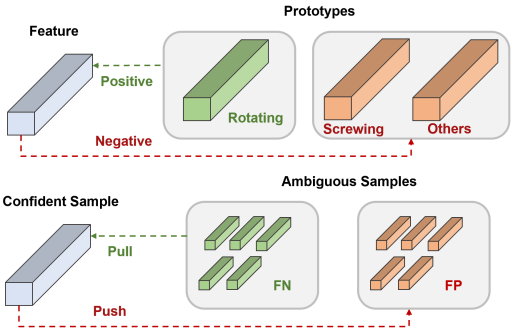


Figure 4: Joint locations in a hand skeleton.



(a) Spatial-Temporal Decoupling



(b) Contrastive Feature Refinement

Figure 5: Feature Refinement Method.

incorporate an independent module for feature refinement within the backbone network. This module decouples hidden feature maps into spatial and temporal components and applies a contrastive learning loss with global class prototypes and ambiguous samples. Notably, the feature refinement module is utilized only during training, without incurring any additional computational or memory costs during inference. In Figure 5, we provide a visual representation of our feature refinement strategies, which we discuss in further detail in subsequent parts.

Multi-Stage Feature Selection To improve the discriminative power of the feature representations learned by the backbone network, we employ a feature refinement (FR) module at three different stages within the network: the 1st, 5th, and 7th layers of the TGN. Each FR module is responsible for refining the hidden features produced by its corresponding stage. At the 5th layer, a strided operation is applied to reduce the spatial resolution of the features. This reduction helps to capture more abstract and higher-level features that are crucial for discriminative representation learning. To refine hidden features, we employ contrastive learning (CL) loss follwing [40]. The CL loss helps optimize the feature representations by encouraging similar samples to be closer in the learned feature space while pushing dissimilar samples apart.

To ensure a balanced contribution from each stage, we introduce a weighting parameter, α_i, for each stage i. These parameters allow us to control the relative importance of each stage in the multi-stage CL loss calculation. The multi-stage *CL* loss, denoted as $\mathscr{L}_{CL}$, is computed as the weighted average sum of the local *CL* losses, $\mathscr{L}^i_{CL}$, calculated by each stage. The formulation for the multi-stage CL loss is given by Equation 1.

$$\mathscr{L}_{CL} = \sum_{i=1}^{3} \alpha_i \cdot \mathscr{L}^i_{CL} \tag{1}$$

Spatial-Temporal Decoupling To address the challenges posed by the complexity of hand joint motion, where coarse-type features can lead to confusion between similar actions with ambiguous spatial appearances or temporal changes, we utilize a spatial-temporal decoupling module. This module aims to simultaneously capture spatial and temporal information to enhance the discriminative ability of hand action representations.

For instance, consider the actions "pick something up" and "put something down." These actions can be easily distinguished using temporal clues. However, when it comes to actions like "attach something to something," more focus on spatial information is required. Therefore, our proposed approach leverages both spatial and temporal cues to accurately differentiate between these actions.

As depicted in Figure 5a, we employ two parallel branches for efficient feature enhancement. Each branch consists of a spatial or

temporal pooling layer, which retains the average value of the related dimension, and a 1×1 convolution layer that compresses the feature into a fixed size. The output features are then flattened to create a unified representation with a channel size of Ch. To further improve the discriminative power, contrastive learning (CL) loss is applied to each branch.

To accomplish the proposed spatial-temporal decoupling feature refinement, we sum the losses from the two branches as follows:

$$\mathscr{L}_{CL}^{i} = CL(f_s^i) + CL(f_t^i) \tag{2}$$

Here, f_s^i stands for the spatial feature, and f_t^i corresponds to the temporal feature vector for stage i. Furthermore, the function $CL(.)$ is responsible for computing the contrastive learning loss using these features. This dual-brach decoupling approach allows us to effectively capture and utilize both spatial and temporal information, enabling a more accurate representation of the actions.

Contrastive Feature Refinement We incorporate contrastive feature refinement into the training of our hand action recognition model, following the approach proposed by [40]. This technique leverages contrastive learning to enhance feature representation at a higher level. Our focus is on addressing the challenge of ambiguous samples by identifying misclassified samples that exhibit similarities to other categories. To overcome this challenge, we gather these misclassified samples and adjust their representations. The adjustments are designed to encourage false negative (FN) samples to be closer to confident samples, reducing the occurrence of false negatives. Simultaneously, we ensure that false positive (FP) samples are pushed away from confident samples, reducing the risk of incorrect classifications. Figure 5b illustrates the process of gathering and adjusting these ambiguous samples at a higher level. By applying contrastive refinement to their features, we enhance the discriminative power and robustness of the learned representation. This technique proves valuable for accurate and reliable action recognition and classification in various domains. For a more detailed understanding of this method, please refer to the supplementary material.

3.2.4 Training Objective

During the training process, our network aims to optimize its performance by minimizing a combined loss function that consists of two components: cross-entropy (CE) loss and a proposed multi-stage contrastive learning (CL) loss.

Cross-entropy loss measures the dissimilarity between predicted probabilities and ground truth labels. It quantifies the inconsistency between the predicted probability scores and the true labels. The CE loss is calculated for each sample in a batch, and the individual losses are averaged to obtain the overall CE loss. The formula for CE loss is as follows:

$$\mathscr{L}_{CE} = -\frac{1}{N}\sum_{i}\sum_{c} y_{ic}\log(p_{ic}) \tag{3}$$

where, N represents the number of samples in a batch. y_{ic} is the one-hot encoded representation of the label for sample i, where $y_{ic} = 1$ only when c is the target class for sample i. The probability score p_{ic} is the predicted likelihood of sample i belonging to class c.

Finally, the CE loss and the previously introduced multi-stage CL loss are combined using a weighted sum to form the full learning objective function as follows:

$$\mathscr{L} = \mathscr{L}_{CE} + \beta \cdot \mathscr{L}_{CL} \tag{4}$$

where $\mathscr{L}_{CE}$ and $\mathscr{L}_{CL}$ are defined in Equations 1 and 3. β is a hyperparameter that supports tuning the relative importance of the CL loss compared to the CE loss, thereby impacting the training process.

3.2.5 Deployment and Inference

In our deployment setup, we chose to host our hand action recognition model on a remote server[1] rather than directly on the MR (mixed reality) headset due to compatibility issues and resource limitations. Many MR headsets are equipped with low-resource specifications, making them unsuitable for hosting complicated deep learning models. Additionally, not all headsets support every machine learning framework, and even if they do, model format conversion can be problematic. By hosting the model on a remote server, we ensured access to ample computational resources and eliminated compatibility concerns. This allowed us to process the incoming hand skeleton data efficiently. We established a TCP connection between the server and the Hololens 2 client app, enabling real-time communication. The client app continuously captured hand skeleton data and transmitted it to the server, where the model was deployed and processed the data in real-time. The predicted action class was then sent back to the client app via the TCP connection, providing seamless feedback and interaction on the MR system. This approach provided a more generalized and universal solution that addressed the compatibility and resource limitations associated with hosting complex models directly on the headset.

4 Experiment

4.1 Case Study: Precision Hand Inspection

MR has found diverse applications across various fields. One particularly important application is its use in industrial settings. In the industrial domain, there is a growing demand for intelligent training and task guidance systems. These systems are based on automatic and real-time understanding of user actions, which heavily involve user hands [5, 8–10]. Therefore, accurate hand action recognition is crucial for the success of such intelligent mixed reality systems.

Given the significance of intelligent MR applications in the industry, we have selected industrial precision inspection tasks [37] as a representative use case to assess the efficacy of our proposed hand action recognition pipeline. These tasks require precise and subtle hand movements, making them an ideal choice to showcase the capabilities of our system. To evaluate our pipeline with the use case, we first build a dataset for hand actions specifically tailored to the hand inspection task by leveraging our developed MR-guided data collection app. This dataset serves as the foundation for training our hand action recognition model. Once trained, we deploy the model and thoroughly test its performance.

Through evaluation of the entire pipeline, including data collection, model training, deployment, and testing, we can determine the efficacy of our approach and its suitability for industrial precision inspection tasks. Additionally, since this use case represents a wide range of domains, the performance of our pipeline will showcase its applicability in other fields as well. Furthermore, the successful implementation of this pipeline demonstrates its potential to be scaled and adapted for any intelligent and adaptive MR system that requires an accurate understanding of hand actions.

4.2 Implementation Details

We developed our MR application for the HoloLens 2 headset, using Unity 2020.3.42f1 and the Mixed Reality Toolkit (MRTK) version 2.7.3. For hand action recognition, we employed PyTorch 1.12.0 as our deep learning framework. The action recognition model was trained and hosted on a single Nvidia GeForce RTX 3080 Laptop GPU. The training process followed the same configuration as described in [40]. In our implementation, we carefully selected hyperparameters to optimize the performance of our methods. Specifically, we set the hyperparameters as follows: $\alpha_1 = 0.1$, $\alpha_2 = 0.2$, $\alpha_3 = 1$,

[1] in a local network

Table 1: Skeleton-Based Hand Action Recognition Datasets.

Dataset	Action Type	Subjects	No. of Joints	No. of classes	No. of sequences	View	Device
SHREC'17 [7]	Gesture	28	22	14/28	2800	Third Person	Intel RealSense
DHG-14/28 [6]	Gesture	20	22	14/28	2800	Third Person	Intel RealSense
FPHA [12]	Social, Office, Kitchen	6	21	45	1175	Ego/FP	Intel RealSense
PHI-16 (Ours)	Hand Inspection	6	26	16	1402	Ego/FP	Microsoft Hololens 2

and $\beta = 0.1$. These values were determined through rigorous experimentation and fine-tuning to achieve the best results in our training dataset.

4.3 Data Collection

Participants Our dataset comprises action sequences collected from a diverse group of 6 participants, consisting of 4 male and 2 female individuals. Among the participants, two had prior familiarity with MR headsets, two had tried them at least once before, and the remaining two had never experienced MR headsets before. Prior to participating, the participants without experience with Hololens 2 went through the Microsoft Tips app to become familiar with basic interactions. Additionally, all participants were provided an introductory video on our data collection app's functionality.

Dataset The dataset encompasses a total of 1,402 action sequences[2], categorized into 16 action classes. The classes were specifically designed to encompass the most common hand actions involved in precision hand inspection [37], which include working with three precision measuring tools (slide caliper, anvil micrometer, and height gage) and two distinct parts (O-ring plug and guide block). Each action class may involve multiple objects, adding complexity to the dataset. Some of the action classes are subtle and closely resemble each other, making the dataset challenging. The action classes are depicted in Figure 6 with representative examples. Table 1 provides a comparison between our dataset and other well-known skeleton-based hand action datasets. Our dataset is unique as it pertains to precision hand inspection and includes the highest recorded number of hand joints (26) compared to existing datasets. Moreover, to our knowledge, it is the only publicly available skeleton-focused hand action dataset collected using an MR headset.

To ensure reliable evaluation and generalization of our proposed methods, we divided the dataset into training and test sets. The training set consists of 892 action sequences, contributed by four subjects. In contrast, the test set comprises 510 action sequences, collected from the remaining two subjects.

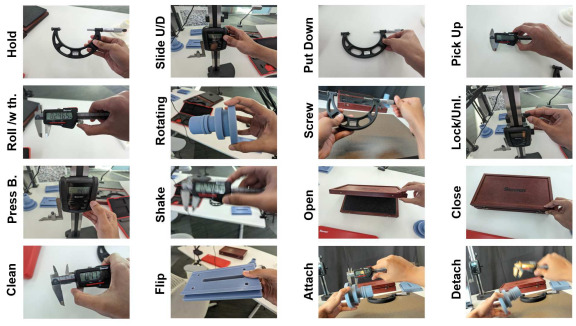


Figure 6: 16 action classes in our hand inspection dataset.

4.4 User Experience Analysis

Here, we present an analysis of the user experience with our MR data collection app based on responses obtained from a questionnaire.

[2]Actions are performed exclusively by the dominant hand, specifically the right hand.

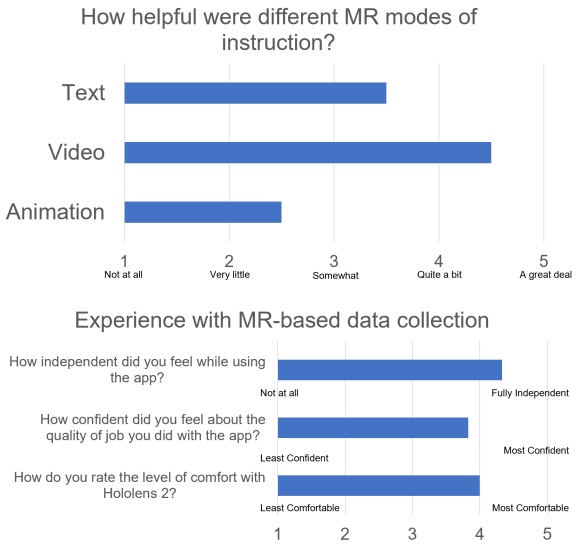


Figure 7: Summary of user survey responses on a 5-point Likert scale.

The questionnaire included a series of questions aimed at gathering feedback on different aspects of the app's usability, specifically focusing on its ability to seamlessly collect high-quality data while ensuring user-friendliness. The responses were collected using a 5-point Likert scale, where 1 represented the lowest level of agreement or satisfaction and 5 represented the highest level. The summarized result is shown in Figure 7, and we discuss it in more detail in the following part, along with its implications.

The participants were asked to evaluate the helpfulness of different modes of instruction in mixed reality (MR), namely text, video, and 3D animation. The findings revealed a clear preference for video instructions, followed by text, while 3D animation was regarded as the least favored. Further exploration through follow-up questions indicated that videos were deemed more effective in providing visual understanding, whereas text alone was considered insufficient. Participants also mentioned that the simplicity of the tasks led them to perform them immediately after watching the videos without relying on the animations. However, for more complex tasks, animations were seen as potentially more useful. Regarding the app's usability, participants rated their level of independence as high, indicating that it was perceived as intuitive and user-friendly, requiring minimal external supervision. When asked about their confidence in the quality of their performance, participants expressed a moderate level of confidence, suggesting that receiving feedback during the task would enhance their assurance and confidence. In terms of comfort with the HoloLens 2 headset, the majority of participants reported a high level of comfort, suggesting that it is ergonomically suitable for conducting long data collection sessions without causing significant discomfort.

In addition to gathering feedback from participants, we conducted interviews and app demonstrations with two experienced experimenters to gain their perspectives on our data collection app. Both

interviewees noted that using the app could significantly decrease their workload during data collection studies, as participants could collect their own data with minimal supervision. They also highlighted the app's ability to facilitate the collection of a larger quantity of high-quality data in less time and with less effort. These insights underscore the potential benefits our app offers experimenters, including increased participant independence, improved data quality, and enhanced research efficiency.

4.5 Hand Action Recognition Model Analysis

In this section, we present a comprehensive evaluation of our proposed hand action recognition architecture. We conduct various experiments to select the optimal backbone network, explore hyperparameters, compare against baseline models, and evaluate real-time performance. The experiments aim to demonstrate the superiority of our approach in terms of accuracy and inference speed for real-time use cases in MR applications.

4.5.1 Backbone Network Selection

To determine the best backbone network for our hand action recognition architecture, we compare four state-of-the-art backbones: 2s-AGCN, TCA-GCN, CTR-GCN, and ST-GCN. We evaluate these backbones individually and with the inclusion of the feature refinement (FR) module. The results are summarized in Table 2.

Upon analyzing the results, we observe that the ST-GCN backbone achieves the highest accuracy and fastest inference speed, making it the most suitable choice for our framework. Furthermore, incorporating the FR module during training improves the accuracy of the ST-GCN backbone. Interestingly, although ST-GCN has more parameters than CTR-GCN, it exhibits significantly higher inference speed. This can be attributed to the complex operations and branches involved in CTR-GCN compared to the relatively simpler structure of ST-GCN, making it more suitable for real-time applications.

Moreover, we notice that the FR module helps the model distinguish between ambiguous actions. Figure 8 illustrates representative examples where our framework without the FR module confuses similar-looking actions, whereas the FR-enhanced model accurately predicts the correct action.

Based on these findings, we select the ST-GCN backbone for further experimentation in our proposed framework.

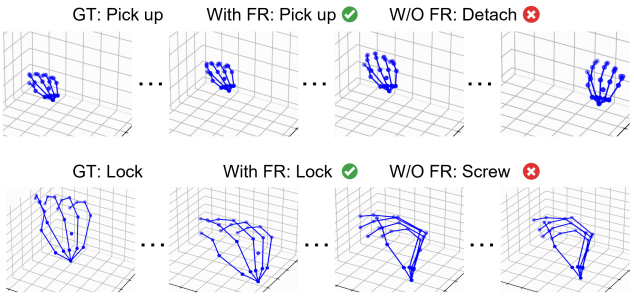


Figure 8: Model with Feature Refinement (FR) module accurately distinguishes similar actions.

4.5.2 Hyperparameter Exploration

During our previous experiments, we conducted the analysis with initial hyperparameters, $\alpha_1 = 0.1$, $\alpha_2 = 0.2$, $\alpha_3 = 1$, $\beta = 0.1$, but these may not be the best fit for our specific use case or dataset. To address this, we decided to explore different hyperparameter configurations in order to enhance our model's performance. The results of our exploration are summarized in Table 3. We initially set $\alpha_3 = 1$, based on prior research [40] putting the most weight on the last stage as it directly influences the final output. With α_3 fixed, we focused on exploring the remaining hyperparameters. By keeping $\alpha_1 = \alpha_2 = 0$ and varying β, we first explored the value of β and discovered that a value of 0.1 yielded the best results. We observed that larger values and very small values for β had a detrimental effect. So, we choose an optimal value in the middle. Subsequently, we proceeded to investigate the impact of α_1 and α_2, experimenting with different combinations to determine the balance of importance between the first and second stages. Our results revealed that assigning a higher weight to previous layers could have a negative influence, suggesting a gradual increase in importance from the early to the final stage resulted in optimal outcomes. We concluded that refining high-level features in the last stage played the most significant role, while low-level features moderately assist in the refinement process. Finally, after thorough exploration, we decided to revert back to our initial hyperparameter configuration of $\alpha_1 = 0.1$, $\alpha_2 = 0.2$, $\alpha_3 = 1$, and $\beta = 0.1$.

Table 2: Performance evaluation of the proposed architecture with different GCN backbones on our hand action dataset.

Backbone	**Params.**	**Acc (%)**	**Inference Speed**
2s-AGCN [27]	2.3M	93.3	260 FPS
+ Refinement	2.7M	93.7	
TCA-GCN [31]	3.2M	92.9	59 FPS
+ Refinement	3.5M	92.8	
CTR-GCN [3]	0.9M	92.8	73 FPS
+ Refinement	1.2M	93.5	
ST-GCN [35]	1.3M	93.5	**352 FPS**
+ Refinement	1.7M	**93.7**	

Table 3: Exploring hyperparameters for the proposed architecture with ST-GCN backbone on our hand action dataset.

α_1	α_2	α_3	β	**Acc (%)**
0	0	1	1	92.3
0	0	1	0.1	92.6
0	0	1	0.01	91.4
1	1	1		92.0
1	0.5	1		92.9
1	0.2	1		93.3
1	0.1	1	**0.1**	92.5
0.5	0.2	1		93.5
0.2	0.2	1		93.5
0.1	**0.2**	**1**		**93.7**

5 Comparison with Baseline Models

In this study, we select two baseline models for comparison. The first model, referred to as FRC-GCN, is a cutting-edge GCN-based model proposed by [40]. Our proposed framework is built upon this state-of-the-art model. The second baseline model is DD-Net, which is a CNN-based model [36]. FRC-GCN is known for its superior accuracy, whereas DD-Net is recognized for its exceptional inference speed. Our primary objective was twofold: to surpass FRC-GCN in terms of accuracy and to outperform DD-Net in terms of inference speed. Ultimately, our aim was to achieve state-of-the-art performance, surpassing both models, to meet the need for real-time action recognition use cases in MR applications, as discussed in Sections 1 and 2. Table 4 presents a comprehensive comparison of our proposed model with the two baseline models, FRC-GCN and DD-Net. Our model achieves state-of-the-art performance, demonstrating an impressive accuracy of 93.7% and an exceptional inference speed of 352 FPS. This indicates that our model outperforms FRC-GCN in

Table 4: Performance comparison of skeleton-based action recognition models on our hand action dataset.

Method	Params.	Acc (%)	Inference Speed
DD-Net [36]	1.8M	75.4	267 FPS
CTR-GCN [3]	1.5M	93.1	45 FPS
FRC-GCN [40] (FR + CTR-GCN)	1.5M	93.3	45 FPS
Ours	**1.3M**	**93.7**	**352 FPS**

terms of accuracy, while utilizing fewer parameters, and exhibits significantly higher inference speed. Furthermore, we surpass DD-Net in terms of inference speed, while also achieving improved accuracy with fewer parameters. These results highlight the robustness of our proposed model and its applicability in real-time scenarios, particularly in mixed reality applications.

In addition to comparing our model with the baselines, we present the results of our model across different action classes using a confusion matrix (Figure 9). The confusion matrix demonstrates the robustness and reliability of our model, as it achieves near-perfect accuracy for most of the action classes. However, despite incorporating a feature refinement method to distinguish ambiguous actions, there are a few instances where our model encounters challenges in correctly classifying similar-looking actions. Specifically, we observe misclassifications where samples of the "holding" action are incorrectly classified as "pressing a button". This misclassification arises from the fact that participants actually press a button while holding a specific gauge, and the subtle finger movement may not be accurately captured by the sensors, leading to the misclassification as "holding". Additionally, there are instances where the "attaching" action is misclassified as "close" due to the similarity in the hand trajectory between attaching a gauge to a part and closing a gauge box lid. These observations indicate that while our model has achieved state-of-the-art performance outperforming other models, there is still room for improvement, particularly in addressing these specific cases.

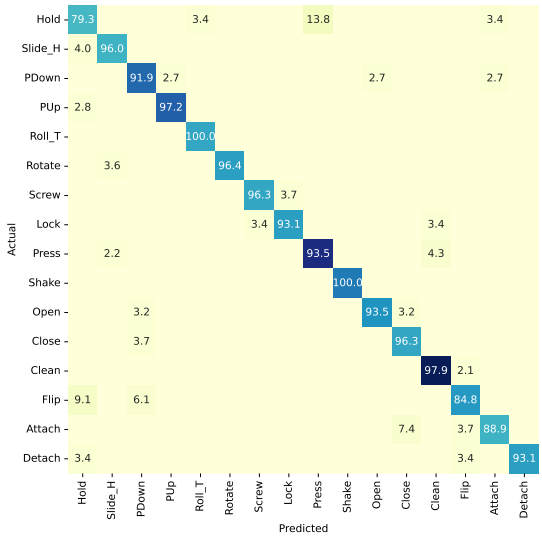


Figure 9: Confusion matrix depicting the performance of the proposed model on our hand action dataset.

5.1 Real-Time Qualitative Analysis

To evaluate the real-time performance of our model, we deploy it in our proposed system and conduct extensive testing. In this real-time setting, we perform various actions and assess whether the model accurately predicts the corresponding actions. Figure 10 shows samples of the real-time testing. During the tests, we observe that our model successfully predicts the correct actions in most cases, demonstrating its robustness in real-time recognition tasks. Although our model achieves a high inference speed of approximately 350 FPS, we notice a slight delay of ~ 1 second when transitioning from one action to another. This delay is primarily due to the temporal window we employed for processing the input data stream, as well as other inherent system latency in the setup. Nevertheless, we anticipate that this slight latency will be acceptable for the majority of mixed reality use cases. Additionally, we acknowledge that our model occasionally generates noisy outputs during action transitions, presenting an area for future improvement. Overall, our proposed model exhibits robustness in real-time hand action recognition, considering the complexity of the task and the intended use case.

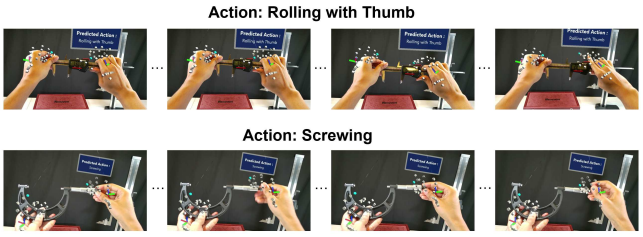


Figure 10: Testing the hand action recognition model in real-time.

6 Conclusion

This paper presents a comprehensive framework for hand action recognition in mixed reality applications. The framework encompasses mixed reality guided data collection, a robust and efficient hand action recognition model, its training and deployment strategies, and testing in a mixed reality system. Through exhaustive experiments and validation, we have demonstrated the effectiveness of our framework, making it suitable for a wide range of intelligent mixed reality use cases. Our mixed reality guided data collection app has been proven to be user-friendly, independent, and efficient in collecting high-quality data. The introduced model is robust, efficient, and highly suitable for real-time applications. The training strategies that we have implemented effectively distinguish between ambiguous hand actions. Additionally, by testing the model in a real-time mixed reality system, we have confirmed its ability to perform well in real-time scenarios.

Despite the success of our framework, we have identified areas for improvement. One such area is the data collection app, where incorporating a feedback mechanism during user action performance would enhance user confidence and assurance. Moreover, in real-time inference testing, we have observed occasional noisy outputs, particularly during transitions between actions. Introducing a mechanism to preserve long-term temporal information and utilize it for real-time action prediction would lead to more stable predictions. Looking ahead, our future plans involve addressing these areas of improvement to further enhance the robustness of our framework. By implementing the suggested enhancements, we aim to make our framework even more reliable and capable of meeting the demands of various mixed reality applications.

REFERENCES

[1] N. Capece, G. Manfredi, V. Macellaro, and P. Carratù. An easy hand gesture recognition system for xr-based collaborative purposes. In *2022 IEEE International Conference on Metrology for Extended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)*, pp. 121–126. IEEE, 2022.

[2] Y. Che, Y. Qi, and Y. Song. Real-time 3d hand gesture based mobile interaction interface. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 228–232. IEEE, 2019.

[3] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13359–13368, 2021.

[4] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 183–192, 2020.

[5] C.-H. Chu and C.-H. Ko. An experimental study on augmented reality assisted manual assembly with occluded components. *Journal of Manufacturing Systems*, 61:685–695, 2021.

[6] Q. De Smedt, H. Wannous, and J.-P. Vandeborre. Skeleton-based dynamic hand gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–9, 2016.

[7] Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. L. Saux, and D. Filliat. 3d hand gesture recognition using a depth and skeletal dataset: Shrec'17 track. In *Proceedings of the Workshop on 3D Object Retrieval*, pp. 33–38, 2017.

[8] J. Dong, Z. Tang, and Q. Zhao. Gesture recognition in augmented reality assisted assembly training. In *Journal of Physics: Conference Series*, vol. 1176, p. 032030. IOP Publishing, 2019.

[9] J. Dong, Z. Xia, and Q. Zhao. Augmented reality assisted assembly training oriented dynamic gesture recognition and prediction. *Applied Sciences*, 11(21):9789, 2021.

[10] W. Fang and J. Hong. Bare-hand gesture occlusion-aware interactive augmented reality assembly. *Journal of Manufacturing Systems*, 65:169–179, 2022.

[11] Q. Gao, Y. Chen, Z. Ju, and Y. Liang. Dynamic hand gesture recognition based on 3d hand pose estimation for human–robot interaction. *IEEE Sensors Journal*, 22(18):17421–17430, 2021.

[12] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 409–419, 2018.

[13] A. Khurshid, R. Grunitzki, R. G. Estrada Leyva, F. Marinho, and B. Matthaus Maia Souto Orlando. Hand gesture recognition for user interaction in augmented reality (ar) experience. In *Virtual, Augmented and Mixed Reality: Design and Development: 14th International Conference, VAMR 2022, Held as Part of the 24th HCI International Conference, HCII 2022, Virtual Event, June 26–July 1, 2022, Proceedings, Part I*, pp. 306–316. Springer, 2022.

[14] T. Kwon, B. Tekin, J. Stühmer, F. Bogo, and M. Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10138–10148, 2021.

[15] J. Lee, M. Lee, D. Lee, and S. Lee. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:2208.10741*, 2022.

[16] J. Liu, X. Wang, C. Wang, Y. Gao, and M. Liu. Temporal decoupling graph convolutional network for skeleton-based gesture recognition. *IEEE Transactions on Multimedia*, 2023.

[17] W. Lu, Z. Tong, and J. Chu. Dynamic hand gesture recognition with leap motion controller. *IEEE Signal Processing Letters*, 23(9):1188–1192, 2016.

[18] Y. Lu and W. W. Mayol-Cuevas. The object at hand: Automated editing for mixed reality video guidance from hand-object interactions. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 90–98. IEEE, 2021.

[19] S. Malik, C. McDonald, and G. Roth. Hand tracking for interactive pattern-based augmented reality. In *Proceedings. International Symposium on Mixed and Augmented Reality*, pp. 117–126. IEEE, 2002.

[20] Y. Murhij and V. Serebrenny. Hand gestures recognition model for augmented reality robotic applications. In *Proceedings of 15th International Conference on Electromechanics and Robotics" Zavalishin's Readings" ER (ZR) 2020, Ufa, Russia, 15–18 April 2020*, pp. 187–196. Springer, 2021.

[21] M. Oberweger and V. Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *Proceedings of the IEEE international conference on computer vision Workshops*, pp. 585–594, 2017.

[22] T. Ohkawa, K. He, F. Sener, T. Hodan, L. Tran, and C. Keskin. Assemblyhands: Towards egocentric activity understanding via 3d hand pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12999–13008, 2023.

[23] A. A. Patil, A. Swaminathan, R. Gayathri, et al. Human action recognition using skeleton features. In *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 289–296. IEEE, 2022.

[24] S.-H. Peng and P.-H. Tsai. An efficient graph convolution network for skeleton-based dynamic hand gesture recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

[25] J. Ryu, D. Kim, and Y. Chai. Corrigible action recognition system through motion-sphere trajectories for standard metaverse actions. In *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 542–547. IEEE, 2022.

[26] A. Sabater, I. Alonso, L. Montesano, and A. C. Murillo. Domain and view-point agnostic hand action recognition. *IEEE Robotics and Automation Letters*, 6(4):7823–7830, 2021.

[27] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12026–12035, 2019.

[28] K. Su, X. Liu, and E. Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9631–9640, 2020.

[29] T. Voillemin, H. Wannous, and J.-P. Vandeborre. Firstpiano: A new egocentric hand action dataset oriented towards augmented reality applications. In *Image Analysis and Processing–ICIAP 2022: 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part III*, pp. 170–181. Springer, 2022.

[30] J. Wang, J. Chen, Y. Qiao, J. Zhou, and Y. Wang. Online gesture recognition algorithm applied to hud based smart driving system. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 289–294. IEEE, 2019.

[31] S. Wang, Y. Zhang, F. Wei, K. Wang, M. Zhao, and Y. Jiang. Skeleton-based action recognition via temporal-channel aggregation. *arXiv preprint arXiv:2205.15936*, 2022.

[32] Z. Wang, Y. Wang, X. Bai, X. Huo, W. He, S. Feng, J. Zhang, Y. Zhang, and J. Zhou. Sharideas: a smart collaborative assembly platform based on augmented reality supporting assembly intention recognition. *The International Journal of Advanced Manufacturing Technology*, 115(1-2):475–486, 2021.

[33] J. Wolf, Q. Lohmeyer, C. Holz, and M. Meboldt. Gaze comes in handy: Predicting and preventing erroneous hand actions in ar-supported manual tasks. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 166–175. IEEE, 2021.

[34] H. Xie, J. Wang, B. Shao, J. Gu, and M. Li. Le-hgr: A lightweight and efficient rgb-based online gesture recognition network for embedded ar devices. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 274–279. IEEE, 2019.

[35] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[36] F. Yang, Y. Wu, S. Sakti, and S. Nakamura. Make skeleton-based action recognition model smaller, faster and better. In *Proceedings of the ACM multimedia asia*, pp. 1–6. 2019.

[37] D. W. Yoo, S. Reza, N. Wilson, K. Jona, and M. Moghaddam. Augment-

ing learning with augmented reality: Exploring the affordances of ar in supporting mastery of complex psychomotor tasks. In *International Society of the Learning Sciences*, 2023.

[38] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Y. Chang, K. M. Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2636–2645, 2018.

[39] X. Zhang, C. Xu, X. Tian, and D. Tao. Graph edge convolutional neural networks for skeleton-based action recognition. *IEEE transactions on neural networks and learning systems*, 31(8):3047–3060, 2019.

[40] H. Zhou, Q. Liu, and Y. Wang. Learning discriminative representations for skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10608–10617, 2023.