Action Recognition for Underwater Gesture Communication in Human Diver and Robot Teaming

Zi-Hao Zhang[1], E. Baker Herrin[1], Jia Guo[2], Aditya Penumarti[1], Zilong He[2], Andres Pulido[1], and Jane Shin[1]

***Abstract*— This paper presents a Spatio-Temporal Transformer-based algorithm for underwater diver hand gesture recognition, forming a key component of diver-robot teaming. Existing computer vision-based approaches primarily rely on frame-wise gesture detection, which often fails to capture motion continuity and suffers under degraded underwater visibility. The presented method integrates temporal modeling to (*i*) improve recognition accuracy by capturing spatio-temporal patterns in hand motion, and (*ii*) increase robustness in challenging underwater environments by leveraging sequential image data, thereby mitigating the impact of intermittent misclassifications. The system is evaluated using real-world underwater footage, demonstrating high recognition accuracy and robustness to lighting fluctuations and partial occlusions. The results highlight the effectiveness and practicality of the presented method for real-world diver-robot collaboration, establishing a foundation for more reliable and intelligent underwater human-robot collaboration.**

I. INTRODUCTION

Recent advancements in autonomous underwater systems have transformed underwater robots from specialized tools into essential collaborators, enabling new capabilities in navigation, exploration, and human-robot teaming for underwater missions [1], [2]. Underwater, hand gestures are the most common form of communication among scuba divers due to environmental constraints. Standardized hand gestures allow divers to convey important messages related to safety, navigation, and task coordination. Extending this gesture-based communication to facilitate underwater human-robot interaction (UHRI) is a natural progression, particularly for close-range interactions where visual communication is most effective. An autonomous system capable of recognizing and responding to diver gestures in real time would significantly enhance the efficiency and safety of diver-robot collaboration in underwater environments.

However, achieving robust recognition of diver hand gestures using computer vision within the human-robot teaming autonomy pipeline presents several challenges. The underwater environment introduces poor lighting conditions, reduced visibility, and dynamic changes in illumination due to water turbidity and refraction. These factors cause inconsistencies in gesture detection and classification inside the autonomy pipeline for human-robot teaming, making traditional static image-based gesture recognition approaches unreliable. In addition, most existing gesture recognition methods focus on classifying hand signals from static images without the temporal component of gestures. Since many scuba diving hand signals involve both static postures and dynamic movements, the temporal modeling can improve the robustness and accuracy of gesture recognition, even with intermittent single-frame misclassifications.

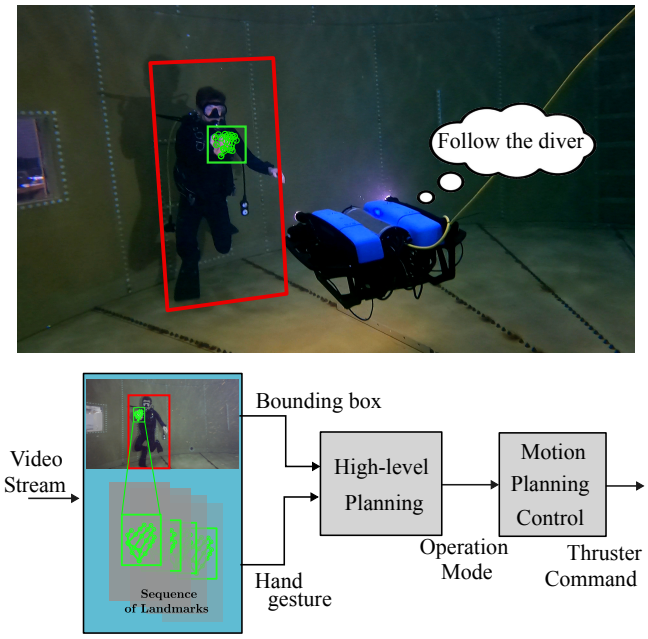


Fig. 1: The presented underwater gesture recognition (blue box) in a human diver-robot teaming autonomy pipeline. The diver is showing a *follow me* hand gesture, which is recognized in real-time by the presented action recognition algorithm. Then, the recognized gesture switches the robot's operation mode into the "diver following mode", which runs a controller that tracks the diver using the detection bounding box.

To address these challenges, this paper proposes a Spatio-Temporal Transformer (ST-TR) framework for underwater gesture recognition in diver-robot teaming. Building on prior Transformer-based approaches for skeleton-based action recognition [3], [4], we introduce a structured encoding of pose and hand keypoints tailored for underwater gestures. Unlike methods that process individual joint embeddings, our model represents the entire joint configuration as a single token per frame, enabling a more lightweight and computationally efficient Transformer architecture. This design

[1]Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA. {zhangzihao, eherrin, apenumarti, andrespulido, jane.shin}@ufl.edu

[2]Department of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14850, USA. {jg2476, zh222}@cornell.edu

prioritizes global frame-level patterns over fine-grained joint correlations, optimizing real-time inference while maintaining robust spatial-temporal modeling for underwater action recognition. Existing datasets for UHRI remain limited, primarily focusing on static gestures [5], [6]. The CADDY dataset [5], for instance, provides stereo image pairs for 16 CADDIAN gestures but lacks labeled sequences for action recognition. In contrast, SDG11 is specifically designed to enable temporal gesture modeling, addressing a critical gap in UHRI and supporting the development of models capable of recognizing dynamic, real-world diver commands.

The main contributions of this paper are as follows:

1) We present the first Spatio-Temporal Transformer model for underwater diver gesture recognition and compare its performance against two alternative approaches: a long short-term memory (LSTM) network that predicts gestures from a sequence of pose keypoints extracted from image frames, and a multi-layer perceptron (MLP) that performs classification based on pose keypoints from a single static frame.
2) We demonstrate that the presented Spatio-Temporal Transformer model effectively captures temporal dependencies, reducing misclassification issues observed in static-image-based approaches.
3) The presented method is validated and demonstrated in real-time in an actual underwater environment with an autonomy pipeline, showcasing its potential for real-time and outdoor implementation in diver-robot teaming applications.
4) We present the first dataset designed explicitly for action recognition of scuba diving hand gestures, following the format of established action recognition datasets such as Kinetics [7].

II. Related Work

Hand gesture recognition is a key component of UHRI, enabling divers to communicate with autonomous underwater vehicles (AUVs). Various approaches have been explored, including wearable sensor-based and vision-based methods. One of the most prominent wearable approaches, developed in the CADDY project, integrates IMU-equipped smart gloves and a stereo camera system on an AUV to improve gesture tracking [5]. Similarly, a multi-sensor fusion system combining dielectric elastomer sensors in gloves with acoustic modems has been proposed for transmitting gesture signals [8], while hydrophone-equipped smart gloves have demonstrated high accuracy in static gesture classification [9]. These glove-based approaches provide robustness and high accuracy; however, in this paper, we consider the scenarios where divers may not have access to specialized gloves or where a camera-only solution is more practical (e.g., confined underwater environment), focusing on vision-based methods. Other hardware setups for UHRI include stereo-based camera for 3D pose estimation [10] and stereo-based diver tracking with sonar sensor [11].

Recent work on vision-only diver gesture recognition eliminates the need for additional wearable hardware, making it more flexible and scalable. In [6], the authors have developed a CNN-based classifier for detecting static gestures from monocular cameras, while handling the case of more dynamic gestures by decomposing them into atomic units. In [12], the authors have validated the robustness of vision-based gesture recognition in underwater environments, demonstrating its applicability for safe human-robot interaction. Their study compared four deep learning-based classifiers with one adapted classical machine learning classifier. In [13], the authors demonstrate that deep learning models such as CNNs and LSTMs consistently outperform traditional computer vision methods in recognizing diver gestures. While some approaches, such as LSTMs, have been explored for dynamic gesture recognition, the majority of existing work has focused on static gesture classification. However, real-world diver gestures often involve dynamic motion sequences, highlighting the need for models that effectively capture temporal dependencies.

Recent research has explored the application of the Transformer architecture for dynamic gesture recognition for robot to robot (R2R) gesture communication [14], stemming from works in multi-robot and diver gesture-based communication [15]. The demonstrated effectiveness of Transformers in capturing spatiotemporal dependencies in the R2R domain motivates the adoption of Spatio-Temporal Transformers for diver gesture recognition in this work as part of the diver-robot interaction pipeline. In the broader domain of action recognition, research in sign language recognition (SLR) provides valuable insights into modeling dynamic hand movements. LSTMs have been shown to be effective for sequential gesture modeling [16], while MediaPipe keypoint tracking has been successfully applied for sign language translation [17], [18]. This work leverages MediaPipe-based tracking in conjunction with a Spatio-Temporal Transformer for underwater gesture recognition. However, while SLR methods assume high contrast and well-lit environments, such conditions are not typically available underwater, necessitating adaptations to handle the challenging visual conditions inherent to the underwater domain.

III. Spatio-Temporal Transformer Action Recognition for Underwater Gesture Communication

This section describes our ST-TR pipeline for scuba hand gesture recognition, consisting of three stages: (1) keypoint extraction via MediaPipe Holistic, (2) input structuring for the Transformer, and (3) gesture classification using an ST-TR model to capture spatiotemporal dynamics.[1]

A. Keypoint Extraction

MediaPipe Holistic is a framework that integrates pose estimation, hand tracking, and facial keypoint detection into a unified model, providing precise skeletal keypoints for human motion analysis [18]. We employ it to extract keypoints

[1]The source code is available at `https://github.com/aprilab-uf/Action-Recognition-for-Underwater-Gesture-Communication-in-Human-Diver-and-Robot-Teaming.git`

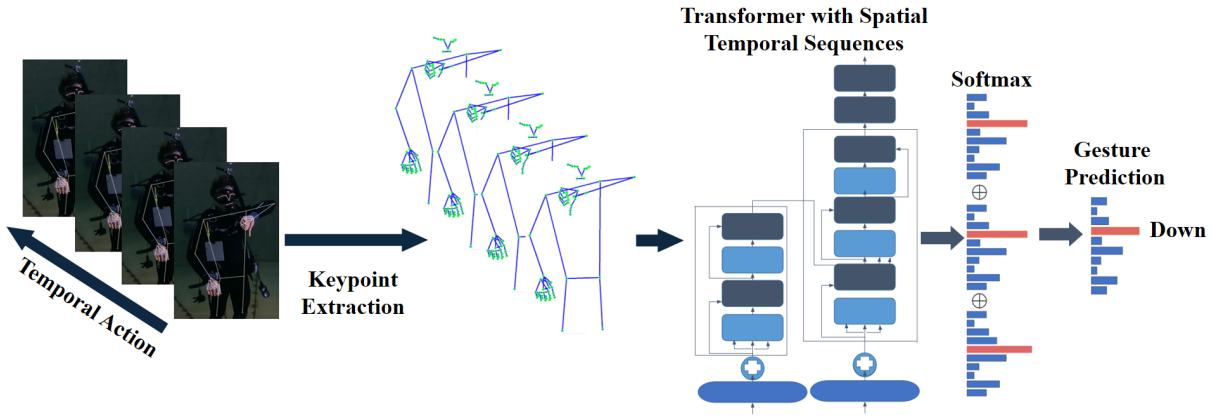
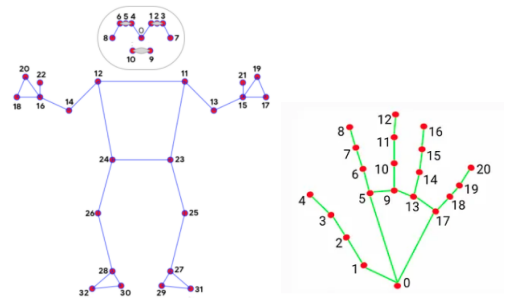


Fig. 2: **Overview of the Transformer-based action recognition method.** A temporal sequence of video frames is processed by a keypoint extraction step, generating skeleton coordinates over time. These coordinates form the input sequence, which is then projected into the model's feature space. Next, temporal encodings are added to preserve positional information across both joints and frames. The resulting tokens are fed into stacked Transformer encoder layers, where multi-head self-attention (button blocks) captures long-range dependencies among frames. After global pooling reduces the sequence to a single embedding, a final classification head (fully connected + softmax) outputs gesture predictions.

for underwater gesture recognition, a crucial step in training our Transformer model.

1) Pose Keypoints: The body pose model in MediaPipe estimates 33 keypoints, covering major joints such as the shoulders, elbows, wrists, hips, knees, and ankles, along with eye, mouth, and ear positions, which can enhance gesture recognition.

2) Hand Keypoints: For hand gesture recognition, MediaPipe tracks 21 keypoints per hand, including the wrist and fingertips, enabling fine-grained motion capture essential for differentiating scuba hand gestures.



(a) (b)

Fig. 3: MediaPipe Holistic (a) pose and (b) hand keypoints. For more details on MediaPipe Holistic implementation and keypoints, see [18]

By combining pose, hand, and facial keypoints, MediaPipe Holistic provides a comprehensive skeletal representation of diver motion, enabling the detection of complex gestures involving both hand and body movements. Each action class is represented as a sequence of keypoints that capture the temporal dynamics of gestures. These keypoints are then normalized and structured into feature tensors, making them suitable for Transformer-based models.

B. Input Structure of the Transformer

1) Keypoint Representation: To train the Transformer model, we first structure the input as a sequence of joint-based features extracted from MediaPipe. Let V be the set of hand joints and T the total number of frames. Each joint $i \in V$ at time t is represented as:

$$\mathbf{J}_{i,t} = \big(x_{i,t}, y_{i,t}, z_{i,t}\big), \quad \forall\, i \in V,\ t \in \{1, \ldots, T\}. \tag{1}$$

where $\mathbf{J}_{i,t}$ denotes the 3D coordinates of joint i at time t. These coordinates are normalized between 0 and 1 for consistency across subjects and videos.

For each frame, the raw coordinates of all N joints are concatenated into a single feature vector, effectively representing the skeleton's configuration at that time step. A sequence of T such feature vectors constitutes the input to the Transformer encoder, allowing the model to capture both spatial and temporal dynamics.

2) Position Embeddings: Since self-attention is permutation-invariant, we must encode the order of the tokens. We incorporate both spatial (joint configuration) and temporal (frame index) embeddings by adding learnable positional vectors to the input features.

Let $\mathbf{u}_{t,j} \in \mathbb{R}^{d_{\text{in}}}$ represent the raw input feature of joint j at time t. We project it to the model dimension as follows:

$$f_{t,j} = W_e \mathbf{u}_{t,j}, \quad W_e \in \mathbb{R}^{d_{\text{in}} \times d_{\text{model}}}. \tag{2}$$

This produces an initial input embedding for each joint configuration at each time step.

C. Transformer Model

1) Multi-Head Self-Attention Mechanism: In our Transformer-based model, each token corresponds to a temporal frame, capturing the features of all 42 joints using their 3D coordinates. The total sequence length, L,

corresponds to the number of temporal frames T, while the model dimension d_{model} represents the flattened set of all joint coordinates per frame. In this work, we use

$$L = T = 60 \text{ (temporal frames)}, \tag{3}$$

$$d_{\text{model}} = N = 42 \times 3 = 126 \text{ (features per frame)}. \tag{4}$$

Each input frame (126 joint coordinates) is projected into query, key, and value vectors via learned linear transformations. Let $X \in \mathbb{R}^{L \times d_{\text{model}}}$ be the input sequence. Then, the attention mechanism computes

$$Q = XW^Q; \quad K = XW^K; \quad V = XW^V \tag{5}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ are learnable weights, and $d_k = d_{\text{model}}/h$ for h attention heads. We use $h = 4$, meaning each attention head operates on $d_k = 31$ dimensions.

Multi-head attention concatenates outputs from the 4 heads and applies an output projection

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{6}$$

This allows the model to learn distinct temporal dependencies across the 60-frame sequences.

2) Residual Connections and Feed-Forward Networks: Each Transformer encoder layer consists of (*i*) Multi-head self-attention (MHA), (*ii*) Residual connection and layer normalization, and (*iii*) Feed-forward network (FFN). After MHA, a residual connection adds the attention output back to the input, preserving raw skeleton features while allowing temporal learning:

$$\text{FFN}(U) = \sigma\big(UW_1 + b_1\big)W_2 + b_2, \tag{7}$$

where $W_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$ and $W_2 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$. This transformation expands and refines joint representations while retaining temporal coherence.

3) Global Pooling and Classification: After processing, the Transformer encoder outputs a sequence of embeddings $\{z_1, z_2, \ldots, z_L\}$, where each z_i corresponds to a specific time step. To obtain a global representation, we apply global average pooling:

$$z_{\text{avg}} = \frac{1}{L}\sum_{i=1}^{L} z_i. \tag{8}$$

This pooled vector is passed through a classification head:

$$y = z_{\text{avg}}W^c + b^c, \tag{9}$$

where $W^c \in \mathbb{R}^{d_{\text{model}} \times C}$ and $C = 11$ (number of gesture classes). Then, a softmax function converts logits into class probabilities, allowing the model to predict the diver's gesture. By leveraging both spatial and temporal dependencies, the Transformer effectively captures complex underwater motion patterns.

IV. Implementation and Experiment Setup

A. Dataset and Implementation Detail

The data collection and real-time demonstration are performed in a 15-foot depth, 26-foot diameter (62,282 gallon) indoor water tank located at the University of Florida. First, the action recognition dataset for underwater scuba gesture recognition presented in this paper (SDG11) is collected from this water tank using BlueROV2, GoPro video camera, and the Raspberry Pi Camera from onboard the BlueROV2. The dataset consists of short scuba hand gesture videos recorded in a controlled water tank environment. The aim of SDG11 is to further the state-of-the-art in UHRI research by providing a robust source of training data for both static and dynamic action recognition algorithms, and provide a standardized set of test data for offline evaluation of diver gesture recognition pipeline performance, assessed across a dynamic set of gestures.[2]

We have selected eleven hand gestures that are used in scuba community, based on their utility for robot teaming. The selected gestures are shown in Table I.

TABLE I: Standardized set of diver hand gestures used in this paper. Each gesture is visually represented with its corresponding label.

For real-time demonstration, we use a BlueROV2 from BlueRobotics in the heavy configuration (eight directional thrusters) to enable full 6-DOF motion. The robot runs Robostack ROS 2 for efficient deployment on minimal hardware [19]. A fiber-optic cable connects the robot to a topside

[2]SDG11: Scuba Gesture Dataset, available at `https://github.com/aprilab-uf/SDG11`, provides annotated recordings of underwater hand gestures for training and evaluating action recognition models in controlled aquatic environments.

desktop computer equipped with an NVIDIA A2000 GPU for real-time inference on the video feed.

We train the Transformer model on an NVIDIA A2000 GPU using a dataset comprising approximately 690,000 trainable parameters, based on hand and body keypoints extracted frame-by-frame via MediaPipe Holistic (v0.10.18). Each input sequence consists of 60 frames (≈2 second), with features stored in .npy format and organized by action class. The dataset includes 11 gesture classes, each with 60 sequences, split into training and testing sets using an 80/20 stratified split to maintain class balance. All features are normalized to zero mean and unit variance.

For baseline comparison, we train an LSTM model under similar conditions with approximately 950,000 trainable and 1,600 non-trainable parameters. We also include a lightweight LeNet-style CNN with only 22,000 parameters. Both models are optimized using cross-entropy loss and the Adam optimizer. To improve robustness, we apply data augmentation during training. Each sequence is jittered with Gaussian noise (μ=0, σ=0.01) to simulate natural skeletal variation, then globally rescaled by a random factor between 0.9 and 1.1 ($\pm$10%) to vary motion amplitude. Temporal distortion is introduced via time warping, where frames are randomly stretched or compressed by 0.9 or 1.1 ($\pm$15%) and interpolated back to a fixed length of 60 frames.

The Transformer architecture includes an input projection layer that maps 126-dimensional keypoint features to a 64-dimensional embedding space, followed by sinusoidal positional encoding. The core consists of four encoder blocks with four self-attention heads, 128 feed-forward units, and a dropout rate of 0.1. Temporal average pooling is applied across the sequence, and the resulting feature is passed to a fully connected classifier that outputs one of 11 action classes. We train the model with a batch size of 64, a learning rate of 1×10^{-3}, and for 100 epochs, resulting in approximately 690,000 trainable parameters.

Our training dataset consists of $C = 11$ action classes, each with 60 video samples. Each video spans 2 seconds at 30 FPS, resulting in 60 frames per sample and a total of 660 samples. We represent each video as a sequence of $T = 60$ tokens, where each token encodes the joint configuration at a specific time step. Each frame is encoded as a 126- or 225-dimensional vector, corresponding to the $\mathbf{J}_{i,t}$ formulation in Eq. (2). Stacking these vectors over time yields a $\mathbb{R}^{60\times126}$ or $\mathbb{R}^{60\times225}$ sequence, which serves as input to the Transformer or LSTM model.

B. Autonomy Implementation in Diver-Robot Teaming

In the diver-robot autonomy pipeline, illustrated in Figure 1, the system deploys low-level stabilization controllers based on feedback from an inertial measurement unit (IMU) to ensure the axial movements are stable and consistent. An additional program is used to initiate a camera stream from the robot to the top-side computer due to resource limitations on the onboard computer. The robot includes a finite state machine (high-level planning) to switch modes dependent on the recognized diver or gesture. The modes deploy four behaviors: "search", "approach", "gesture recognition", and "stop". The "search" mode includes turning in place to search for a diver. When a diver is detected inside the FOV, the robot switches into the "approach" mode, where the robot approaches the diver using a feedback controller based on the bounding box center and size with respect to the frame.

Once the robot detects that it is close enough to the diver, it switches into "gesture recognition" mode, which has several subroutines for the gestures given in Table I. These include simple wrench commands and feedback controllers to move according to the divers' commands. The operation modes are illustrated in Figure 4 as an example scenario.

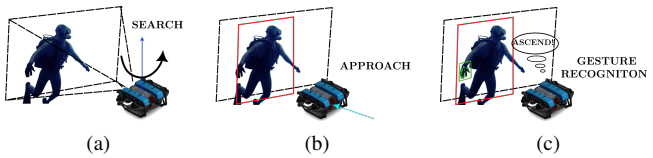


Fig. 4: Different operation modes of the autonomy pipeline for the robot-diver gesture recognition for the robot. For example, this figure describes one scenario where the robot switches its mode from (a) search to (b) approach, and to (c) gesture recognition.

C. Evaluation Procedure and Metrics

We evaluate the accuracy and robustness of gesture recognition using recorded diver gesture videos in the presented dataset, SDG11. We compare the performance of our ST-TR-based model against two baseline approaches: **(*i*) a static baseline** that classifies gestures from single-frame image data; and **(*ii*) an action recognition baseline** that performs sequence-based gesture recognition using a sequence of keypoint data.

For (i) the static baseline, we construct a LeNet-style CNN to perform gesture recognition from static image frames. In parallel, we adapt the MediaPipe hand gesture recognizer by leveraging its MLP model provided through MediaPipe Model Maker, serving as an additional baseline for static image–based gesture recognition. The gesture recognizer comprises of a hand keypoint estimator and a gesture classifier. . The gesture classifier contains an embedding model and a classification model, both of which are fully connected neural networks. The customization only involves training of the classifier model. Compared to the ST-TR model introduced in this paper, the baseline model is trained on the dataset of images sliced from SDG11 videos. The image dataset contains 57661 images which are categorized into the classes listed in Table I. The number of images of each gesture class ranges from 3754 to 6130. Besides, a *None* class is also included for the gestures other than the listed ones. The training/testing ratio of the baseline model is 80/20. The confusion matrix of the baseline model is shown in Figure 6.

For (*ii*) action recognition baseline, we implemented an LSTM-based model, which is trained on the same diver

gesture video data. Performance is evaluated by finding the confusion matrix and multi-class F1-score averaged across 5 stochastic runs of each respective model, to account for variability in model performance.

V. Results

A. Action Recognition Performance Comparison

The confusion matrix for our ST-TR model is visualized in Figure 5 (c)-(d), while Figure 6 and Figure 5 (a)-(b) visualize the confusion matrix from the baselines: MLP, LeNet-style CNN and LSTM models, respectively. The confusion matrices highlight key performance differences among the models. The MLP model (Figure 6) exhibits a strong diagonal but contains critical failure modes, as indicated by dark off-diagonal elements. The LeNet-style CNN model shows strong classification performance, with most gestures. Some confusion remains among similar gestures such as ASEND, DESCEND, and STOP, which show notable off-diagonal misclassifications. The LSTM model (Figure 5 (a)-(b)) produces more scattered misclassifications without major failures, while the ST-TR model (Figure 5(c)-(d)) has a slightly weaker diagonal than the MLP but avoids critical misclassifications, demonstrating more consistent performance.

As shown in Figure 5(c)-(d), the Transformer trained with MediaPipe Holistic keypoints generally outperforms the hand-only approach on dynamic gestures (e.g., *buddy-up*), largely because the Holistic extraction includes the forearm and can still track arm movement even when the hand itself is not detected. For gestures that rely heavily on precise hand motion, such as *follow me*, the hand-only approach can sometimes excel—provided the hands are tracked accurately.

Meanwhile, the Transformer's attention mechanism tends to consistently distinguish subtle underwater gestures, reaching about 93.6% accuracy under Holistic tracking. We do note, however, that the hand model occasionally misclassified actions like *right* as *me* or *left*, likely due to limited movement cues and a relatively low keypoint extraction rate. In contrast, in Figure 5(a)-(b), the LSTM underperforms on smaller datasets, lacking sufficient spatiotemporal modeling capacity. As a whole, these observations confirm that a broader body context (Holistic) plus attention-based modeling better captures dynamic movements than a purely hand-centered pipeline, especially in challenging underwater settings.

B. Performance Analysis based on Gesture types

A comparison of the confusion matrices in Figure 6 and Figure 5(c)-(d) highlights the strengths and limitations of each approach, based on different gesture types. Here, we refer to the hand gestures that rely on hand motions as "dynamic" gestures; on the other hand, we refer to the hand gestures that convey information based on the pose of the hands as "static" gestures. For example, *buddy-up* is a dynamic gesture, while *okay* is a static gesture.

Specifically, the performance comparison for dynamic gestures is summarized in Table II. For *buddy-up* gesture, the ST-TR-based model achieves 100% accuracy, surpassing the static baseline (Figure 6), which attains 96.3%, a 3.7% improvement. Except for *follow-me* gesture, the presented ST-TR-based method works better than the baseline methods in recognizing dynamic gesture.

TABLE II: Comparison of precision (*TP / (TP + FP)*) for specific gestures of interest across all three models, tested using keypoints from MediaPipe Hands. Best score in bold.

	Gestures				
Model	**Buddy Up**	**Follow Me**	**Left**	**Right**	**You**
MLP	96.3%	**97.8**%	52.0%	0.0%	31.0%
LSTM	58.3%	8.3%	41.7%	33.3%	8.3%
ST-TR	**100.0**%	95.8	**95.8**%	**95.8**%	**83.3**%
LeNet-5	69.2%	33.1%	69.2%	66.1%	71.2%

TABLE III: Comparison of Weighted F1 Scores

Model	Weighted F1 (average)
LSTM (Hand)	0.088
LSTM (Holistic)	0.076
Transformer (Hand)	0.459
Transformer (Holistic)	0.757

On the other hand, for static gestures such as *descend*, the static model achieves 100% accuracy, while the Transformer-based approach performs worse, reaching only 87.5%. This discrepancy suggests that frame-level classifiers like the static baseline are more effective at recognizing gestures with minimal motion, whereas the ST-TR model excels at capturing temporal dependencies in dynamic gestures. These results indicate that while our Transformer-based approach improves recognition for highly dynamic actions, static models remain advantageous for gestures with limited movement.

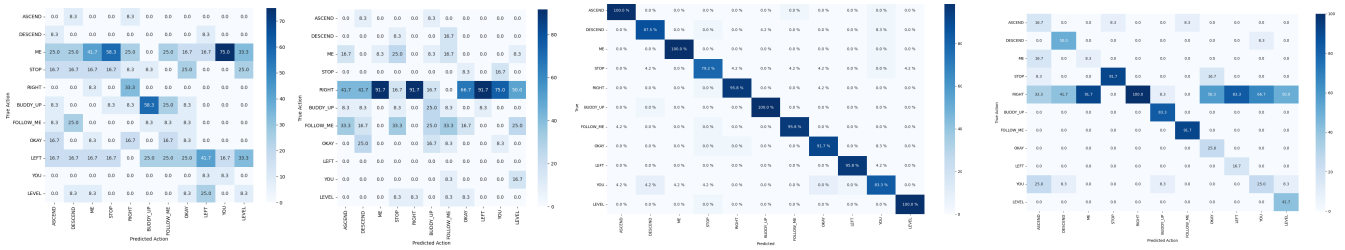
C. Robustness in Gesture Recognition

Experiments show that static image-based methods are less robust in diver-robot teaming due to inconsistent classification. As illustrated in Figure 7, the MLP baseline intermittently detects the correct gesture (*buddy up*) across frames. From randomly selected *buddy up* videos totaling 1095 frames, the MLP predicts *None* for 63.53

D. Ablation Study

We compare two keypoint extraction approaches, MediaPipe Holistic and MediaPipe Hands, combined with either LSTM or Transformer-based classifiers. As shown in Table III, the Transformer consistently outperforms LSTM across all settings, validating the strength of attention mechanisms in modeling long-range temporal dependencies for underwater gesture recognition.

Among all combinations, the Holistic with Transformer model achieves the highest weighted F1 score (0.757). This is primarily due to its higher frame-level keypoint extraction rate ($\approx$75%), enabled by the use of upper-body references that improve robustness under poor visibility. In contrast, while the Hand model yields highly accurate keypoints, it detects them on only $\approx$35% of frames, limiting overall sequence-level performance.



(a) (b) (c) (d)

Fig. 5: Confusion matrices of LSTM and ST-TR (ours) models trained with MediaPipe Holistic or Hands model: (a) LSTM+Holistic (b) LSTM+Hands (c) Ours+Holistic (d) Ours+Hands, following the same convention as Figure 6.

(a) (b)

Fig. 6: Confusion matrix of precision for ***(i)* the static baseline** MLP model provided by MediaPipe Model Maker (a) and the LeNet-style CNN model (b). The x-axis represents predicted classes, and the y-axis represents true classes for the 11 scuba gestures from Table I. Each column shows the percentage distribution per class, with correct classifications along the diagonal. Darker blue squares indicate higher percentages.

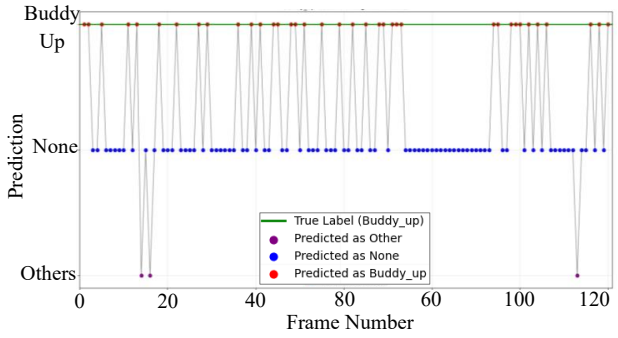


Fig. 7: Inconsistent correct classification of the MLP (static image-based gesture recognition) baseline over image frames from a *buddy up* gesture video.

Figure 8 highlights a typical failure case of the Holistic model: occlusion or background blending (e.g., an arm overlapping the torso) causes incorrect full-body pose predictions, which in turn corrupt hand keypoint estimates. The Hand model avoids this failure by focusing solely on visible hand regions, allowing more reliable tracking even when full-body pose estimation fails.

These results suggest that the Holistic model benefits from broader context but is vulnerable to full-body estimation errors, while the Hand model is more precise but suffers from low frame coverage. The combination of Holistic features with Transformer-based temporal modeling strikes the best balance for robust gesture recognition in underwater environments.

(a) (b)

Fig. 8: Comparison of keypoint generation for diver in case of ambigious diver pose for MediaPipe Holistic (a) and MediaPipe Hands (b). Note that in (a) MediaPipe Holistic attempts to fit pose keypoints to the Diver's right arm incorrectly, leading to incorrect hand keypoints.

E. Real-time Implementation in the Autonomy Pipeline

While the proposed method is validated in a controlled underwater setting, real-time deployment faces hardware challenges. The onboard camera captures at 15 FPS, while the Transformer performs best at 30 FPS. The model is also too large for untethered deployment; running it on an A2000 GPU leads to system overload. To address this, we implement the ST-TR model on an NVIDIA Jetson Orin Nano and achieve real-time inference on prerecorded underwater videos. Future work includes mounting the edge device on the ROV for in-situ operation.

F. Limitations

While the ST-TR model demonstrates strong overall performance, it underperforms compared to the MLP model on static gestures, such as *descend*. This result indicates that, for real-world deployment, it may be beneficial to combine the strengths of both ST-TR (ours) and MLP models to more effectively handle mixed static–dynamic gesture recognition

tasks. Additionally, the dataset used to train the ST-TR model is limited to indoor underwater environments, which constrains its generalizability. The challenges remain, including sensitivity to underwater lighting, turbidity, and backscatter. To enhance robustness, the dataset should be extended to include outdoor settings with diverse lighting and visibility conditions. Moreover, this study evaluates the system in a controlled tank environment, but testing in open water introduces additional challenges such as currents, variable visibility, and occlusions from bubbles and glare. Finally, the current system does not support untethered deployment, as the BlueROV still relies on a tethered connection to an external computer or edge device for executing the action recognition task.

VI. Conclusion and Future Work

This paper presents a Spatio-Temporal Transformer-based (ST-TR) action recognition pipeline for underwater diver-robot communication using hand gestures. Unlike prior methods focused on static recognition, the proposed ST-TR model captures spatio-temporal patterns in diver signals, enabling more accurate recognition of dynamic hand movements. Experimental results demonstrate improved recognition accuracy across several gestures, including *Me, Right, Buddy Up, Left, You* and *Level*, outperforming both static and traditional dynamic gesture models. The system's real-time integration on a BlueROV2 and the introduction of a new underwater gesture dataset further validate the feasibility of deploying gesture-based diver-robot communication in practice.

Future resesarch will focus on expanding the gesture vocabulary to support multi-gesture sequences for higher-level intent inference, as well as enabling bidirectional communication through robot-to-diver feedback mechanisms. In addition, real-time image enhancement and segmentation techniques [20]–[22] will be investigated to improve feature extraction and visual robustness, further advancing capabilities for underwater scientific exploration, commercial diving, and autonomous operations.

ACKNOWLEDGMENT

The authors would like to thank Quang Pham and Eli Heskin for their work on CNN comparison and Detectron2-based pose estimation. We also extend our gratitude to Daniele Fragiacomo and Silvia Ferrari for their valuable contributions to the ideation of this work.

References

[1] A. Paradise, S. Surve, J. C. Menezes, M. Gupta, V. Bisht, K. R. Jang, C. Liu, S. Qiu, J. Dong, J. Shin, *et al.*, "Realthasc—a cyber-physical XR testbed for AI-supported real-time human autonomous systems collaborations," *Frontiers in Virtual Reality*, vol. 4, p. 1210211, 2023.

[2] S. Surve, J. Guo, J. C. Menezes, C. Tate, Y. Jin, J. Walker, and S. Ferrari, "Unrealthasc – a cyber-physical XR testbed for underwater real-time human autonomous systems collaboration," in *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, 2024, pp. 196–203.

[3] J.-H. Song, K. Kong, and S.-J. Kang, "Dynamic hand gesture recognition using improved spatio-temporal graph convolutional network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6227–6239, 2022.

[4] C. Plizzari, M. Cannici, and M. Matteucci, "Spatial temporal transformer network for skeleton-based action recognition," *arXiv preprint arXiv:2012.06399*, 2020. [Online]. Available: https://arxiv.org/abs/2012.06399

[5] A. G. Chavez, A. Ranieri, D. Chiarella, E. Zereik, A. Babić, and A. Birk, "CADDY Underwater Stereo-Vision Dataset for Human–Robot Interaction (HRI) in the Context of Diver Activities," *Journal of Marine Science and Engineering*, vol. 7, no. 1, p. 16, 2019. [Online]. Available: https://doi.org/10.3390/jmse7010016

[6] M. J. Islam, M. Ho, and J. Sattar, "Understanding human motion and gestures for underwater human–robot collaboration," *Journal of Field Robotics*, vol. 36, no. 5, pp. 851–873, 2019. [Online]. Available: https://doi.org/10.1002/rob.21837

[7] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[8] D. Nad, C. Walker, I. Kvasić, D. O. Antillon, N. Mišković, I. Anderson, and I. Lončar, "Towards advancing diver-robot interaction capabilities," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 199–204, 2019. [Online]. Available: https://doi.org/10.1016/j.ifacol.2019.12.307

[9] D. W. O. Antillon, C. R. Walker, S. Rosset, and I. A. Anderson, "Glove-based hand gesture recognition for diver communication," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 9874–9886, 2023.

[10] D. T. Kutzke, A. Wariar, and J. Sattar, "Autonomous robotic realignment for face-to-face underwater human-robot interaction*," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5920–5926.

[11] D. Nad, C. Walker, I. Kvasić, D. W. O. Antillon, N. Mišković, I. A. Anderson, and I. Loncar, "Towards advancing diver-robot interaction capabilities," *IFAC-PapersOnLine*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:212876553

[12] A. G. Chavez, A. Ranieri, D. Chiarella, and A. Birk, "Underwater vision-based gesture recognition: A robustness validation for safe human–robot interaction," *IEEE Robotics & Automation Magazine*, vol. 28, no. 3, pp. 67–78, 2021.

[13] M. A. Martija, J. I. Dumbrique, and P. Naval, "Underwater Gesture Recognition Using Classical Computer Vision and Deep Learning Techniques," *Mathematics Faculty Publications*, Mar. 2020.

[14] S. S. Enan, M. Fulton, and J. Sattar, "Robotic detection of a human-comprehensible gestural language for underwater multi-human-robot collaboration," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3085–3092.

[15] M. Fulton, C. Edge, and J. Sattar, "Robot communication via motion: Closing the underwater human-robot interaction loop," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4660–4666.

[16] S. Kausar and M. Y. Javed, "A Survey on Sign Language Recognition," in *2011 Frontiers of Information Technology*, Dec. 2011, pp. 95–98.

[17] B. Duy Khuat, D. Thai Phung, H. Thi Thu Pham, A. Ngoc Bui, and S. Tung Ngo, "Vietnamese sign language detection using mediapipe," in *Proceedings of the 2021 10th International Conference on Software and Computer Applications*, ser. ICSCA '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 162–165. [Online]. Available: https://doi.org/10.1145/3457784.3457810

[18] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, *et al.*, "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.

[19] T. Fischer, W. Vollprecht, S. Traversaro, S. Yen, C. Herrero, and M. Milford, "A RoboStack Tutorial: Using the Robot Operating System Alongside the Conda and Jupyter Data Science Ecosystems," *IEEE Robotics & Automation Magazine*, vol. 29, no. 2, pp. 65–74, June 2022, conference Name: IEEE Robotics & Automation Magazine. [Online]. Available: https://ieeexplore.ieee.org/document/9646255

[20] C. Li, C. Guo, W. Ren, R. Cong, J. Hou, S. Kwong, and D. Tao, "An underwater image enhancement benchmark dataset and beyond," *IEEE transactions on image processing*, vol. 29, pp. 4376–4389, 2019.

[21] M. J. Islam, Y. Xia, and J. Sattar, "Fast underwater image enhancement for improved visual perception," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 2, pp. 3227–3234, 2020.

[22] M. Jeon and S. Lee, "Segmentation of respiratory bubbles in underwater diver image using pixel coordinate information and k-means clustering," in *2024 21st International Conference on Ubiquitous Robots (UR)*, 2024, pp. 695–700.