TransTCNet: Transformer-Based Temporal-Contextual Network for Low-Latency Typing Interfaces on Edge Devices

Asif Ullah , Zhendong Song * , Waqar Riaz , Yizhi Shao , Xiaozhi Qi

Posted Date: 21 April 2026

doi: 10.20944/preprints202604.1498.v1

Keywords: surface electromyography (sEMG); typing recognition; deep learning temporal-contextual modeling; human-computer interaction; real-time neural interfaces

 doi:10.20944/preprints202604.1498.v1

Article

TransTCNet: Transformer-Based Temporal-Contextual Network for Low-Latency Typing Interfaces on Edge Devices

Asif Ullah [1,2], Zhendong Song [2,*], Waqar Riaz [3], Yizhi Shao [2] and Xiaozhi Qi [2]

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

[2] Institute of Ultrasonic Technology, Shenzhen Polytechnic University, Shenzhen 518055, China.

[3] Nanfang College, Guangzhou, Guangzhou, P. R China

* Correspondence: szdeer@szpu.edu.cn

Abstract

A distinct typing interface using surface electromyography (sEMG) can facilitate silent, hands-free typing by interpreting muscle activity in relation to specific keystrokes. Character-level recognition poses challenges compared to the recognition of unseemly gestures, due to insensitivity to slight temporal variations and the fusion of muscle dynamics. Temporal Feature is vital, since when typing, there may be irrelevant dissimilarities in how people press keys, and even in body movements that coincide. This paper proposes TransTCNet, a two-stage deep neural network design with a causal convolutional layer to learn local features and a transformer-based component to learn long-range temporal interactions. We evaluated our network on a publicly available 26-class typing sEMG dataset acquired from 19 individuals. The model achieved a validation accuracy of 96.53%, exceeding the baseline models. Our study revealed generalization among participants, and the AUC values were also high (>0.994) across all classes. The model was significantly reliable and displayed high prediction confidence (>0.9), enabling us to obtain a high training accuracy rate (97.86%) for real-time filtering decisions. TransTCNet could be suitable for wearable and edge devices due to its efficient architecture and low inference cost. The model's ability to consistently decode fine-grained neuromuscular signals across users makes it a suitable choice for real-time applications such as adaptive user interfaces, virtual and augmented reality, prosthetic control, and communication systems.

Keywords: surface electromyography (sEMG); typing recognition; deep learning temporal-contextual modeling; human-computer interaction; real-time neural interfaces

1. Introduction

Effective interaction with complex computing systems requires high-throughput, intuitive channels to capture user intent. Even though conventional peripherals like keyboards, mice, and touchscreens are universal and well-established, new ideas in virtual, augmented, and mixed reality require more seamless, immersive interfaces that minimize the need for physical peripherals [1]. It can also be difficult to connect and communicate with physically disabled people using a standard gadget. Neurotechnology enables the direct decoding of user intent from neurological and neuromuscular cues, offering a promising approach to address these problems [2].

One of the many modalities of the sensory system is surface electromyography (sEMG). It is a non-invasive method for accurately measuring the electrical activity produced by skeletal muscles. sEMG was originally intended to control prosthetics [3] and has been used as a variety of sensing tools. More applications are being implemented for gesture recognition [4–6], facial expression analysis [7–9], handwriting replication [10,11], and, most recently, character-level typing recognition

 doi:10.20944/preprints202604.1498.v1

[12]. Such developments may have serious effects on human-computer interaction (HCI) when applied to problems where conventional input mechanisms do not work or cause interference.

One engaging application of sEMG is typing activities. The resulting sEMG patterns of every single keystroke during typing are frequently complex and delicate due to the highly fine motor activity. It is not simple to group and identify patterns when individual letter keys (A-Z) are treated as separate categories [12]. This issue would solve the further development of systems of silent communication and sEMG-regulated virtual keyboards. Both technologies would be applicable across a broad spectrum of assistive technologies, immersive environments, and wearable devices.

Another field that has seen the application of sEMG-based activity recognition systems has a positive impact on healthcare. Such systems can assist with physical therapy, track motor function, and enable control of myoelectric prostheses by clarifying muscle function. Machine learning is making these systems dynamic signal processors rather than mere signal processors [13–15]. Deep learning has revolutionized the field, eliminating the need to design features manually, and end-to-end spatiotemporal patterns can be learned from raw sensor data [16–19]. Deep learning models can also be effectively deployed in real-world systems, as they can adapt to user, sensor malfunctions, and environmental changes through continuous optimization.

Several machine learning and deep learning techniques have been used to classify human activities from sEMG signals [20]. Al-Qaness et al. proposed Multi-ResNet, a multi-resolution residual network that employs attention mechanisms alongside residual neural networks to automate feature extraction from sensor data. The performance of this strategy is considerably better than that of the traditional HAR benchmark [21]. Similarly, Hnoohom et al. proposed an Att-BiLSTM network consisting of a convolutional and recurrent network with attention mechanisms that enable the recognition of spatial and temporal dependencies [22]. Choudhury et al. applied ensemble and shallow learning models to smartphone-collected data to achieve competitive accuracy levels [23,24]. These studies emphasize the growing diversity and sophistication of deep learning models used to simulate complex biosignals.

Nonetheless, most extant research continues to have many fundamental boundaries. Primarily, sEMG signals are sensitive to sensor placement, muscle fatigue, physiological variations among individuals, and generalization across sessions, which is a recurring issue [25,26]. Secondly, with respect to fine-grained categorization, such as the typewriter of alphabetes, several models appear to be over-tuned to simple or static gestures [27,28]. Third, deep networks are prone to overfitting, especially when using small datasets [29]. Lastly, attempts to interpret decision boundaries and represent cognitive structure in a physiologically meaningful way have been minimal, thereby constraining interpretability [30].

This study aims to address gaps in current classification techniques for EMG signals obtained during keyboard use by leveraging finer-grained feature sets. There are publicly available bilateral forearm EMG recordings from 19 subjects performing 26 different keyboard presses in two different sessions [12]. Such recordings contain 16 channels of EMG data along with synchronized records of keyboard presses, thus allowing many possibilities for evaluating different evaluation strategies across subjects and sessions. The objective of this study is to create a highly accurate/reliable classification model of keyboard presses with the intent of allowing for the creation of typing systems that interface with the sEMG derived from the movement of the user's forearm.

Therefore, we present the TransTCNet, a transformer-based model for generating temporal-contextual neural networks that have learned short-term (temporal) and long-range sequential relationships from all EMG data generated by typing. By using attention-based encoding and dilated causal convolution, this model leverages long-range context across all input signals while quickly identifying localized patterns. The method, as described in detail in **Figure 1**, shows that our model was built specifically to classify fine-grained, character-level signals from the surface electromyography (sEMG) system (as opposed to some previous studies). It achieves optimal performance by recording 16 channels of bilateral forearm muscle activity at low latency and high

 doi:10.20944/preprints202604.1498.v1

frequency. We have demonstrated that this is the first model to utilize temporal attention and convolutional layers specifically for typing interfaces.

On a collection of 26 typing classes, our method outperformed alternative baseline models (e.g., 1D-CNN, CNN-SE, SE-TCN) with a maximum validation accuracy of 96.53%. Our findings indicate that temporal contextual models, such as those developed by the authors, have significant potential to enable users with disabilities to type adaptively using hands-free, wearable, or immersive devices.

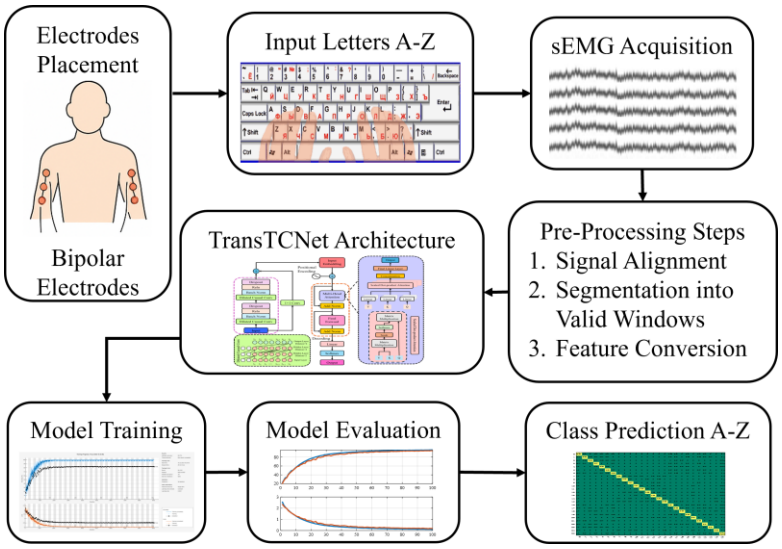


Figure 1. End-to-end pipeline for sEMG-based character-level typing recognition using a temporal-contextual deep learning framework.

2. Materials and Methods

2.1. Dataset Description

2.1.1. Keyboard Typing sEMG Dataset

A publicly available surface electromyography dataset for character-level keyboard typing identification was used in this study [12]. Fine-grained neuromuscular data captured during key presses enables future research on silent text entry, adaptive interfaces, and bioelectric control systems.

Participants performed a structured typing task using a standard QWERTY keyboard, with a unique class for each letter of the alphabet (A-Z), creating a 26-class classification problem. The study used a scheme to provide finger-to-key mapping, ensuring that participants maintained a stable typing posture throughout the typing trial.

The bilateral sEMGs (two forearms) were captured at 2000 Hz and bandpass filtered from 10 to 500 Hz using a 4th-order Butterworth filter to reduce motion artifact and high-frequency noise while preserving the frequency content of interest for motor control. Sixteen bipolar electrodes (8 on each arm) were used to collect the raw signals.

For every letter trial, the five presses of the spacebar before the trial served as an alignment mechanism so that each test trial had previously pressed the spacebar. All letter key presses were also recorded with a metronome set to 75 BPM, ensuring temporal consistency across subjects and repetitions. This pacing allowed participants to reduce mental workload and muscle fatigue, thereby improving the correspondence between sEMG spikes and keystrokes. This was an essential feature for constructing supervised models from data. However, the fixed timing used in the study will not account for inherent individual typing rhythms or the effects of fatigue from prolonged typing. Both factors significantly influence sEMG characteristics during normal daily tasks. Future studies could

 doi:10.20944/preprints202604.1498.v1

adopt free-typing paradigms and fatigue-related data to improve ecological validity and generalizability further.

The proposed methodology was completed for two repetitions of each letter in each subject testing session. Consequently, each letter had 20 repetitions per subject, resulting from completing the task in two separate sessions on two separate days, and remained for analysis. It is important to note that all keypress trials and related subject-key activities, i.e., background and idle sEMG, were temporally segmented. Consequently, only completed active-character-class (A-Z) data were available for analyses and could not include any "rest" class identification. Future work will require continuous sEMG acquisition to detect both the active class and the rest state, enhancing the capability for real-time applications.

2.1.2. Participants

The dataset consists of 19 healthy individuals (5 men, 14 women) with a mean age of 31 ± 7 years. Ethical approval for the study protocol was granted by the University Health Network Research Ethics Board (Protocol #21-6137). Each subject participated in two recording sessions (T1 and T2), during which all alphabetic characters were recorded in trials. The experimental process enables the robust validity of session-based scenarios.

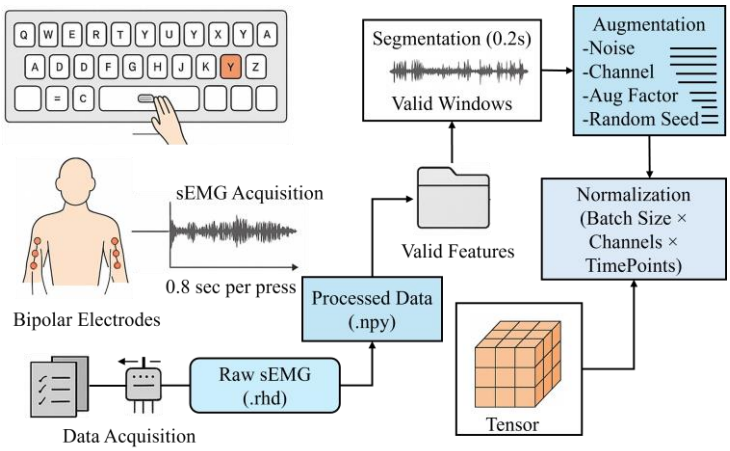
2.1.3. Data Structure

The data was structured in a way that made it easy to train models and conduct cross-session assessments. Raw sEMG signals were first recorded in proprietary electrophysiological formats and synchronized with the keystroke events, along with respective timestamp logs. These synchronized signals were then preprocessed and converted to standard NumPy arrays to enable effective processing in the subsequent machine learning pipeline.

The dataset was divided into subject-specific and subdivided into session-specific directories for structured access. Archiving pre-segmented 0.2-second signal windows that were temporally aligned to each keypress was executed so that they could be used in a window-based recognition. Besides, session-wise and user-wise feature sets were disclosed to facilitate generalization analysis, including intra-subject and inter-session generalization.

2.2. Data Preprocessing

The overall data-preprocessing process (as shown in Figure 2) comprises the basic steps for processing raw sEMG data into normalized tensors, which can then be used as input to the model. Before the classification dataset is prepared for training the model, there are five main phases of pre-training data preparation: data curation, signal segmentation, signal augmentation, signal normalization, and signal reshaping.



 doi:10.20944/preprints202604.1498.v1

Figure 2. Overview of the sEMG preprocessing pipeline: from raw signal acquisition during alphabetic typing to segmentation, augmentation, normalization, and tensor formatting for model input.

2.2.1. Valid Session and Window Selection

The first phase of data preparation involved determining and preserving whole, valid sessions (in which all 26 alphabetic classes were recorded correctly) and deleting trials that contained synchronization errors (or had no label), thus preserving an intact dataset. The signal windows for sEMG data were set to 0.20 seconds and aligned with the timing of the keypress onsets. The segmentation method used ensured a high degree of temporal proximity between neuromuscular activity and the corresponding keypress gestures. Thus, classification performance will benefit from this segmentation method. The selection of a 0.20-second window length was ultimately supported by an analysis of classification performance as a trade-off between accurate gesture classification and response time. Classification accuracy increased slightly with increasing window length; however, beyond a 0.20-second window length, accuracy decreased. Also, because real-time systems require response window lengths of less than 0.30 seconds to support typical typing speeds (180+ keystrokes per minute), the use of a 0.20-second window length provides a compromise between providing high fidelity recognition and a low latency inference, all of which are critical for applications in the neural interface domain.

2.2.2. Data Augmentation

To improve model generalization and preserve the physiological significance of the surface EMG samples, all original sample windows (n=3) were triplicated by creating 2 synthetic copies. The original and synthetic sample windows were created by applying a bandpass filter (50-450 Hz) to all channels and by simulating small-amplitude/transient noise by adding 0-mean controlled Gaussian noise with s = 0.01 to each sample. Additionally, 1 of the 8 channels was assigned a value of 0 for each original/synthetic sample to simulate a temporary sensor dropout or electrode disconnection. These distortions were deliberately kept small to prevent the integration of spectral artifacts that might alter underlying neuromuscular activity. The random seed was fixed to ensure reproducibility of training operations. This conservative augmentation approach enhances stability against small acquisition artifacts (such as motion artifacts or impedance changes) and is physiologically realistic. Nevertheless, the protocol doesn't currently model spatial perturbations, including electrode shift, which is a significant source of performance loss in practice and should be validated using specific strategies in further investigation.

2.2.3. Normalization and Tensor Formation

To ensure stability throughout the training process and a homogeneous contribution from all features, each window was normalized per channel to have zero mean and unit variance. The resulting normalized signals were then converted into a 3D array of size (batch size × channels × timepoints), thus making them compatible with the deep learning structure. High-resolution sEMG recognition depends on efficient learning of both local temporal variations and long-range sequence patterns that support tensor representation.

2.3. Neural Network Architecture

2.3.1. TransTCNet Pipeline

TransTCNet is a temporal-contextual deep learning network designed to efficiently and accurately detect fine-grained electromyographic (sEMG) signals during keyboard use. This technique examines multi-channel time-varying signals, primarily recording short-time variations and thereby storing a detailed sequence structure. As seen in Figure 3, the model encompasses two major steps:

 doi:10.20944/preprints202604.1498.v1

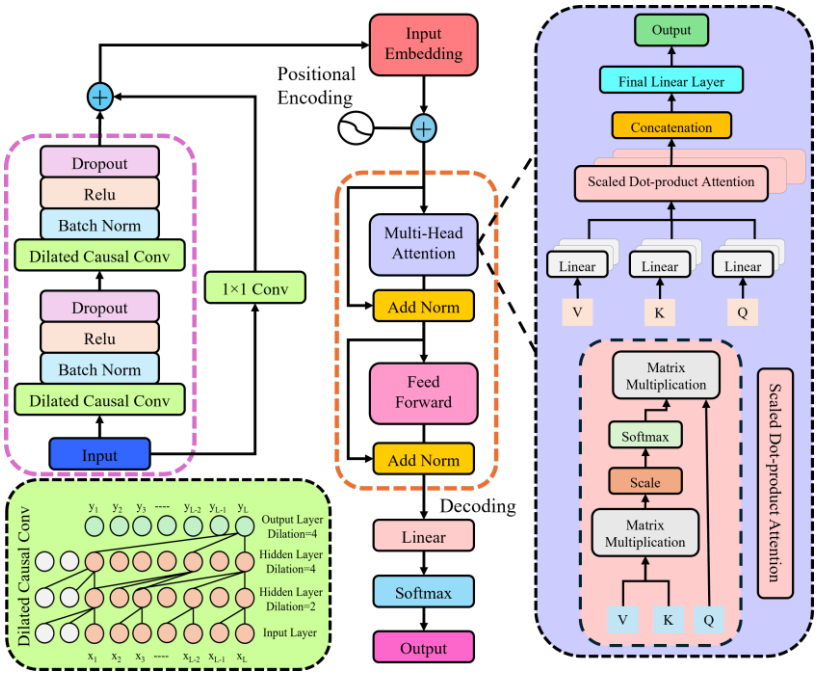


Figure 3. Architectural overview of the proposed TransTCNet framework, highlighting temporal feature extraction via dilated causal convolutions and contextual encoding using multi-head self-attention.

2.3.1.1. Local Temporal Feature Encoding

The first part of TransTCNet focuses on extracting localized temporal signals of raw sEMG signals by exploiting dilated causal convolutions. This approach ensures that the model assesses signal sequences in the inherent temporal sequence whilst allowing the receptive field to deepen without augmenting network depth.

The input to the network is a 3D tensor X ∈ RB×C×T, where B is the batch size, C is the Number of EMG channels (e.g., 16), and T is the number of time samples per segment (e.g., 400). The data may be normalized and reshaped for 1D convolutional operations.

Short-range dependencies are extracted with the help of dilated causal convolutions. This operation ensures that the output at any given time depends only on the current and past inputs. Mathematically, the dilated convolution is represented as:

$$y(t) = \sum_{i=0}^{k-1} w(i) \cdot x(t - d \cdot i) \qquad (1)$$

where k is the kernel size, d is the dilation factor, and w(i) are learnable weights.

Each temporal block incorporates batch normalization to stabilize activations, ReLU activation for non-linearity, and Dropout for regularization, along with residual connections to preserve gradient flow. When the input and output dimensions differ, a 1×1convolution is used to align them, which can be illustrated as

$$X_{res} = Conv1 \times 1(X) \qquad (2)$$

This alignment is necessary because using dilated causal convolutions with varying dilation factors can modify the effective receptive field and, in some cases, lead to inconsistencies between input and output channel dimensions. The 1×1 convolution acts as a learnable linear projection that preserves temporal resolution while matching the number of feature channels, thus enabling residual addition between layers. This operation ensures both shape compatibility and stable gradient propagation within the temporal block.

The output of the temporal block is denoted as:

$$Z_{temp} \in R^{B \times C' \times T} \qquad (3)$$

where C′ is the number of output channels (e.g., 64), and T may be reduced due to padding and dilation control. This feature map is then passed to global contextual sequence modeling.

 doi:10.20944/preprints202604.1498.v1

2.3.1.2. Global Contextual Sequence Modeling

After capturing local dependencies, the next stage of the model focuses on capturing long-range contextual information across the entire temporal sequence, which is critical for recognizing typing patterns distributed over time. First, the output tensor from the previous stage is reshaped to match the expected input format for the attention mechanism. The tensor, denoted as Xin ∈ RB×C′×T, where B is the batch size, C′ represents the number of output channels, and T denotes sequence length (time steps), is permuted to Xin ∈ RB×T×C′, where each time step represents a feature vector t.

Next, a linear projection maps the feature vectors to a unified model dimension, dmodel, a hyperparameter that defines the model's feature space (e.g., 128). This projection is mathematically expressed as:

$$E = X_{proj} = X_{in} \cdot W_{proj} + b_{proj} \in R^{B \times T \times d_{model}} \quad (4)$$

where Xproj is the projection matrix, and bproj is the bias term. This operation results in the projected tensor E ∈ RB×T×dmodel, which is now in a form suitable for further attention-based processing.

Since attention mechanisms are permutation-invariant, meaning they do not consider the temporal order of the sequence, positional encoding is added to inject information about the relative or absolute positions of the time steps in the sequence. The positional encoding P is either learnable or sinusoidal, and it is added to the projected tensor E as follows:

$$E_{pos} = E + P \quad (5)$$

where P ∈ R1×T×dmodel represents the positional encoding matrix, which allows the model to account for the temporal order of the sequence.

Afterward, a multi-head self-attention (MHA) layer is applied to enable the model to attend to multiple positions in the input sequence simultaneously, each step being represented by a different "head (h)." For each attention head, the mechanism computes a weighted sum of the input sequence, where the weights are derived from a query (Q), a key (K), and a value (V). Mathematically, the attention mechanism for a single head is defined as:

$$Attention(Q, K, V) = Softmax(\frac{QK^{\top}}{\sqrt{d_k}})V \quad (6)$$

where Q, K, V are linear projections of the input Epos and dk is the dimensionality of keys (usually dk=dmodel/h). The result of this computation is a weighted sum of the values, which is the output of one attention head. This operation helps capture dependencies between different positions (time steps) in the sequence by computing attention scores based on the similarity between the queries and keys.

For multiple heads (denoted by h), the results from each head are concatenated, and a final linear projection is applied to get the final multi-head output. The operation is mathematically expressed as:

$$MultiHead(X) = Concat(head_1, \ldots, head_h) \cdot W^O \quad (7)$$

where each headi is the output from a single attention head, and WO is a learned weight matrix that projects the concatenated result back into the desired output dimension. This operation enables the model to capture temporal dependencies by assigning attention weights based on the similarity between time steps. Each head focuses on different aspects of the sequence, and their concatenation allows integration of diverse temporal patterns into a unified representation.

Next, layer normalization is applied to stabilize the learning process. It is achieved by normalizing the MHA layer's output, adding it back to the input, and using a residual connection to preserve gradient flow during backpropagation. Mathematically,

$$Z_1 = LayerNorm(E_{pos} + MultiHead(E_{pos})) \quad (8)$$

Following this, a position-wise feedforward network (FNN) is applied independently at each time step in the sequence. The FFN consists of two linear layers with a ReLU activation function in between. The mathematical operation is:

$$FFN(z) = max(0, z \cdot W_1 + b_1) \cdot W_2 + b_2 \quad (9)$$

 doi:10.20944/preprints202604.1498.v1

where W1, W2, and b1, b2 are the learned weight matrices and bias terms for the two linear layers. The ReLU activation introduces non-linearity to the network, enabling it to learn complex representations.

After the FFN layer, layer normalization is again applied to the output of the FFN and added to the input (residual connection), ensuring stability during training:

$$Z_2 = LayerNorm\big(Z_1 + FFN(Z_1)\big) \quad (10)$$

After repeating this process for N layers (e.g., N=4), we obtain a final tensor designated as,

$$Z_{final} \in R^{B\times T\times d_{model}} \quad (11)$$

To convert this sequence into a fixed-size representation, temporal average pooling is applied to average the embeddings across all time steps, effectively summarizing the entire sequence into a single vector:

$$F_{agg} = \frac{1}{T}\sum_{t=1}^{T} Z_{final}[:, t, :] \quad (12)$$

Finally, the aggregated features are passed through a linear classifier to obtain the output logits for each of the 26 alphabet classes. The output is computed as:

$$y_{logits} = F_{agg} \cdot W_{cls} + b_{cls} \in R^{B\times 26} \quad (13)$$

where Wcls and bcls are the weight matrix and bias term for the final linear layer. The result is a 26-dimensional vector of logits, which is subsequently passed through a softmax function to obtain the final probability distribution over the classes.

This two-stage design—combining local temporal feature extraction with long-range contextual modeling—enables TransTCNet to capture the intricate muscle activation sequences required for fine-grained sEMG-based character-level typing tasks. The model is designed to generalize well across different subjects and sessions, making it suitable for real-world applications such as muscle-driven text entry and assistive technologies.

Algorithm: TransTCNet summarizes the overall learning pipeline and describes the processes used during training and inference with the proposed model. The process begins with local temporal pattern extraction via dilated causal convolutions, followed by global context encoding via attention mechanisms. Training is based on the cross-entropy loss function and has been improved using the Adam optimizer. During the evaluation phase of the proposed model, accuracy and other performance metrics are calculated from the predicted probabilities.

Algorithm: TransTCNet Training and Inference

Input: Preprocessed sEMG windows X, class labels y, learning rate η, number of epochs N, batch size B

Output: Trained model parameters θ*, evaluation metrics

1. Initialize TransTCNet parameters θ
2. Initialize optimizer (Adam) and loss function (Cross-Entropy)
3. for epoch ← 1 to N, do
4. for each minibatch $(X_b, y_b) \in D_{train}$, do
5. $Z_t \leftarrow$ DilatedCausalConv1D (X_b)
6. $Z_c \leftarrow$ Self Attention (Z_t)
7. $\hat{y} \leftarrow$ Softmax (Classifier (Z_c))
8. $\mathcal{L} \leftarrow$ Cross Entropy $(\hat{y}, y_b)$
9. $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$

end

10. Model Evaluation (accuracy, F1-score, etc.)

end

Return: Final trained model θ*, performance metrics

3. Experimental Setup

3.1. Training and Validation Split

 doi:10.20944/preprints202604.1498.v1

The data was stratified so that 80% was used for training and 20% for validation, ensuring that both subsets had the same class distributions. The validation set was created by splitting the sessions so that the model could be evaluated on data it did not see during training, from those same subjects at a different time. Each fold was generated randomly, but the random number generator used a fixed seed value to replicate training results.

3.2. Hyperparameters and Evaluation Metrics

The optimal set of hyperparameters found during the experiments to train the TransTCNet model is shown in Table 1, with configurations that will converge the model optimally and generalize well. In addition to using a batch size of 32 and a learning rate of 1e-4, the model was trained for 100 epochs using the cross-entropy loss function with the Adam optimizer. The hyperparameters were determined based on earlier experimental results and discussions with experts who have performed similar time-series classification tasks. In addition, the training process was very efficient, with the overall training completed in 90.6 minutes (1.51 hr), which is twice as fast as the previous iterations of the model using the same data and NVIDIA GeForce RTX 5080 GPU. The training time was reduced due to the effective operation of the data-loading pipelines and the proper execution of the dilated causal convolutional functions in the model architecture. Finally, the model's evaluation consisted of numerous performance metrics (total accuracy, confusion matrix analysis, precision, recall, F1-score, and area under the curve (AUC)) that assess how well the model classifies objects, including resilience to class imbalance and error patterns. The detailed results of the above metrics and analyses, including both intra-session (96.53%) and inter-participant (87.57%) performance analyses, are provided in the Results section.

Table 1. Training Hyperparameters and Computational Settings used for the TransTCNet model.

Hyperparameter	Value
Batch Size	32
Learning Rate	1×10^{-4}
Optimizer	Adam (β_1=0.9, β_2=0.999, $\varepsilon=10^{-8}$)
Epochs	100
Augmentation Factor	3
Train/Validation Split	80:20
Model Architecture	Transformer Encoder
Input Channels	16
Embedding Dimension (d_model)	128
Attention Heads	8
Transformer Layers	4
Dropout Rate	0.1
Positional Encoding	Learned
Feedforward Dimension	512
Loss Function	Cross-Entropy
Gradient Clipping	1.0
Weight Initialization	Xavier Uniform
Model Parameters	849,818
Training Time	1.51 hours

3.3. Hardware/Software

All tests were performed on a high-performance computer equipped with an NVIDIA RTX 5080 (12 GB VRAM) graphics card, an Intel Core i9-275HX processor, and 32 GB of DDR5 RAM, running Linux—Python 3.13.2 as the leading development platform and PyTorch 2.8.0 as the basic deep learning framework. The supported libraries were NumPy 2.0.0, SciPy 1.14.0, scikit-learn 1.6.0,

 doi:10.20944/preprints202604.1498.v1

Pandas 2.2.2, and Matplotlib 3.9.0, which were used for data management, statistical analysis, and visualization. Subsequent processing, performance evaluation, and visualization of the results were done in MATLAB R2024a. The entire pipeline was trained in a Conda-based virtual environment to ensure the library's compatibility and reproducibility. Both PyTorch and NumPy use a fixed random seed (seed=42) to ensure reproducible training results across iterations. There were a few significant optimizations that were implemented, such as Multi-process data loading (4 workers), pinning memory, Dynamic gradient clipping (max norm=1.0), Adaptive learning rate scheduling (ReduceLROnPlateau with factor=0.5, patience=10), an efficient data augmentation pipeline (Gaussian noise (s=0.01)), and Memory-efficient embedding extraction, which is done during validation. The training pipeline took 16-channel sEMG data at 2 kHz, in each 0.2-second window (400 timepoints), normalized (per channel) to zero mean, unit variance, and then passed through the model.

4. Results

4.1. Accuracy and Loss Curves

Figure 4 shows the training and validation accuracy of the TransTCNet model of 100 epochs on the 26-class keyboard typing sEMG dataset. Subplot (a) shows the accuracy progression; both the training and validation curves improve steadily throughout training. The training accuracy peaks at epoch 100 (97.98%), the validation accuracy peaks at epoch 93 (96.53%), and the final accuracy at epoch 100 is 96.02%. The high similarity between training and validation accuracy, with an end-to-end gap of just 1.96%, indicates that generalization with minimal overfitting occurred. Subplot (b) shows the loss curve. The loss curve declines sharply in the first 20 epochs and then gradually approaches 0.0663 by epoch 100. The same occurs with the validation loss, which also stabilizes at 0.1773. These two loss curves differ slightly, indicating that the model generalizes well. Training was completed in 90.6, indicating it can potentially be deployed on edge devices. Additionally, TransTCNet has been shown to learn unique temporal characteristics of sEMG signals from specific key presses and continues to perform well on previously unseen validation data.

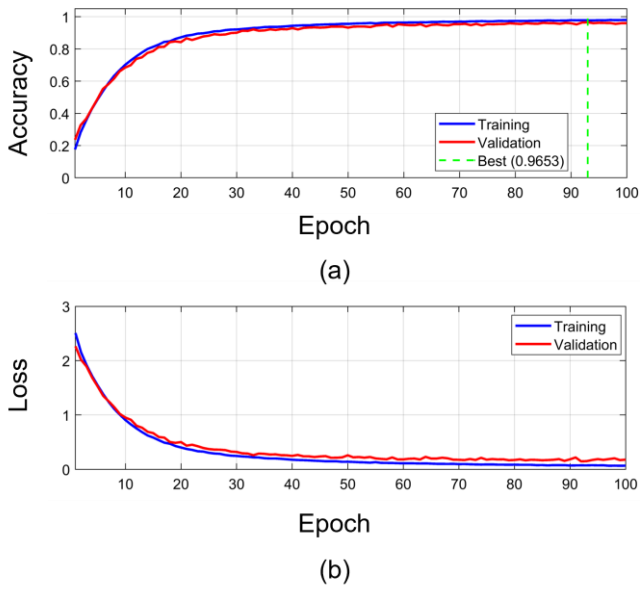


Figure 4. Training and validation curves of the TransTCNet model. (a) Accuracy progression over 100 epochs and (b) Cross-entropy loss convergence during training and validation.

4.2. Confusion Matrix Analysis

 doi:10.20944/preprints202604.1498.v1

The normalized confusion matrices of the 26-class sEMG typing classification task of TransTCNet are illustrated in Figure 5 and by computing the mean of each class in the confusion. High diagonal dominance was achieved across all classes, with 84.07% - 92.14% accuracy (Classes E - P, respectively) for each class. Recognition rates > 87 were achieved for nearly every class, and across the alphabet, performance was comparable. Several important observations about the error analysis were found. Most confusion occurs among biomechanically similar key presses: there is significant confusion between J → N and E → D, with E → D being the most frequently misclassified pair. The majority of misclassifications occur when an adjacent finger presses a key or when two keys recruit the same muscle group. The classes that had lower diagonal values (E, B, and C) indicate that they are more difficult to differentiate because the muscle activation patterns for these classes may be less distinct and/or will vary from press to press. The overall accuracy based on the aggregated confusion matrices was 87.83%, and the class standard deviation across the classification was 1.98%. Thus, this provides evidence that they can accurately identify fine-grained typing gestures and shows which character pairs may benefit from targeted training and/or feature refinement in upcoming versions of the algorithm.

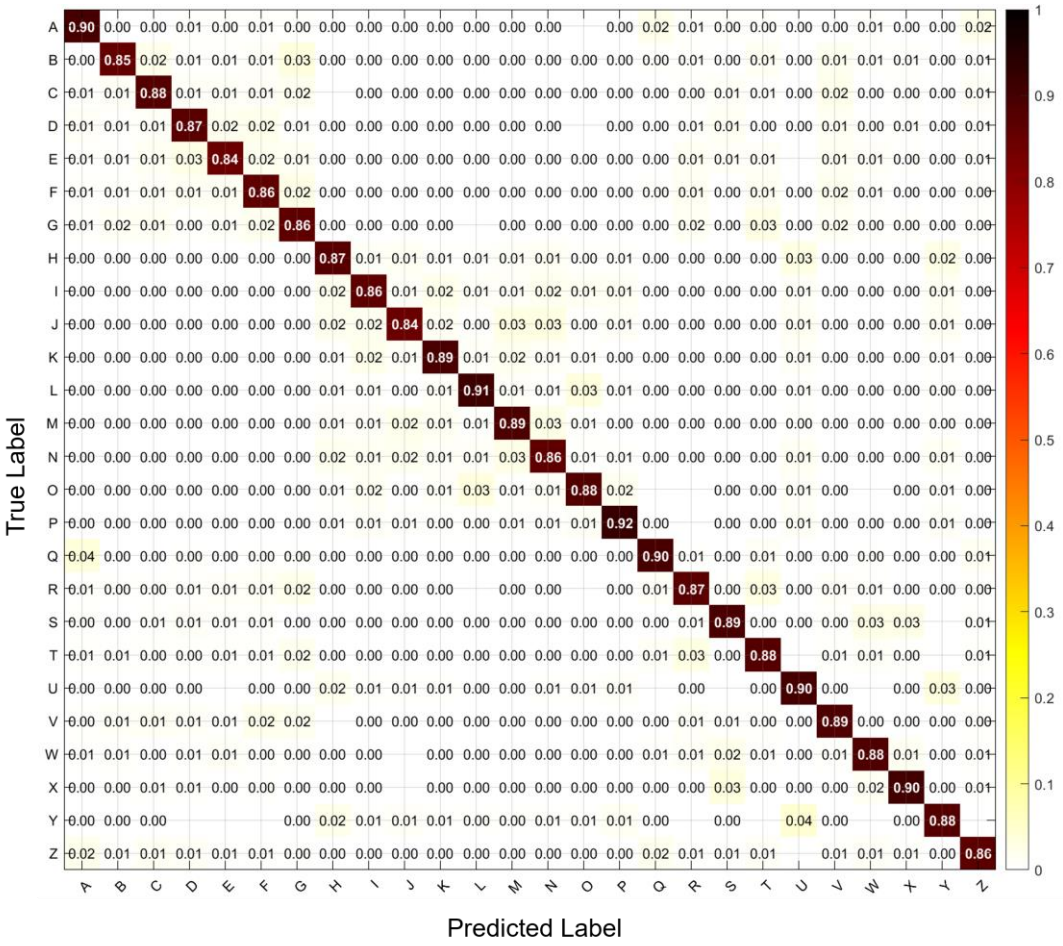


Figure 5. Normalized confusion matrix for the 26-class sEMG typing classification task using the TransTCNet model.

4.3. Participant-Wise Performance Analysis

Figure 6 compares the accuracy of each participant in performing a typing task involving 26 character classes across 19 participants as a whole and shows the accuracy for each participant's overall typing ability. The first part of the figure depicts a bar graph depicting each individual's overall accuracy, showing the average overall accuracy for each individual (87.57%) and showing the range of individuals' overall accuracies (i.e., the lowest overall accuracy measured was 87.30% and

 doi:10.20944/preprints202604.1498.v1

the highest was 87.81%). All participants' individual accuracies are very close to the overall average, so the standard deviation (0.51%) of participants' overall accuracy is very small, indicating that participants' overall performance was highly consistent. The second part of the figure depicts the distribution of participants' individual accuracies relative to the mean and median (87.60%). The standard deviation of participants' individual accuracies (0.13%) is very small, indicating that inter-subject physiological variation is not strongly correlated with overall accuracy in sEMG-based interfaces. The consistency in participants' performance across P1-P19 supports the generalizability of TransTCNet, requiring little personalization or calibration for individual users. The findings indicate cross-subject applicability and successful use of the results with all participants (i.e., a wide range of individuals with minimal decreased performance due to individual differences in muscle physiology, electrode position, and typing biomechanics).

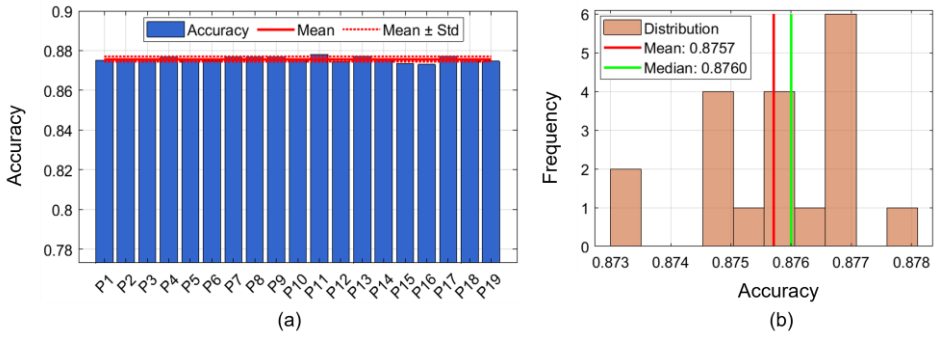


Figure 6. Participant-wise performance analysis. (a) Bar chart showing individual participant classification accuracy across 19 participants. (b) Distribution histogram of participant accuracies, with mean and median values indicated.

4.4. Class Level Performance

Figure 7 shows four visualizations that support the performance metrics of the 26 alphabetical classes. The values of precision, recall, and F1-score are shown in subplot (a), and most classes are in the 85%-92% range. This balanced performance across the three metrics indicates stable classification performance, without significant bias toward false positives or false negatives. The subplot (b) results indicate that the correlation between the precision and recall values is highly positive, with correlation coefficients of more than 0.95, and the values are closely grouped around the diagonal. The classes that have the highest F1-scores (above 0.90) are P, L, and X, and those that have a slightly lower performance (approximately 0.85) are E and B.

The subplot (c) demonstrates the distribution of F1-scores for classes with slight variance, with the mean and median near 88%. The histogram indicates that most classes exhibit performance within a narrow range, suggesting consistent uniform recognition strength rather than occasional brilliance. The relationship between class accuracy and sample frequency is examined in subplot (d), which shows no significant correlation between the two variables. This implies that classification performance remains stable in the dataset, supporting the model's robustness to class imbalance. Combined, these visualizations demonstrate that TransTCNet is well capable of classifying fine-grained typing gestures and exhibits balanced performance across all 26 character classes.

 doi:10.20944/preprints202604.1498.v1

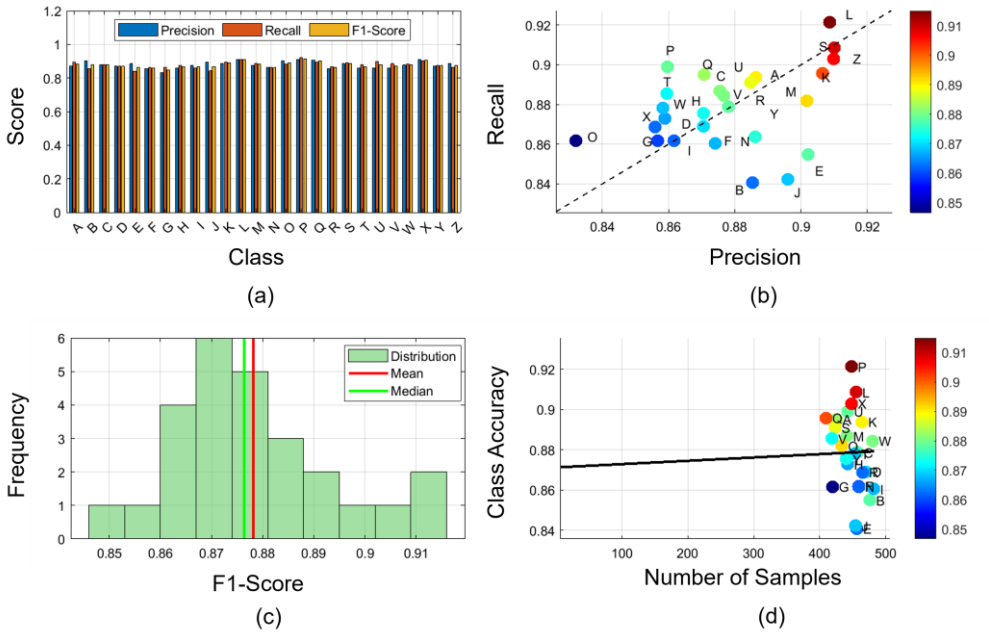


Figure 7. Class-level performance analysis across 26 alphabetical classes. (a) Precision, recall, and F1-score metrics for each character. (b) Scatter plot of precision versus recall with color-coded F1-scores. (c) Distribution histogram of F1-scores across classes. (d) Relationship between class accuracy and sample frequency.

4.5. Feature Space Visualization

Figure 8 shows dimensionality reduction representations of the learned feature embeddings. The results of PCA projections are presented in panel (a), and t-SNE transformations in panel (b), both of which project high-dimensional sEMG signal representations into two-dimensional spaces to facilitate interpretation. Most alphabetical classes exhibit distinct clusters, particularly those with distinctive muscle activation patterns, such as A, L, and X. However, there is apparent interference between biomechanically similar keypress pairs, such as E-D, J-N, and B-G, consistent with the confusion patterns identified in Section 4.2. Such neighboring clusters indicate that the model acquires representations that preserve the anatomical and neuromuscular associations among similar typing gestures. The t-SNE shows smaller, more segregated clusters than the PCA plot, indicating that nonlinear manifold learning better represents fine-scale discriminative structure in the feature space. Classification performance is related to the quality of separation: higher-accuracy classes are defined by well-isolated clusters, whereas frequently confused character pairs are located in overlapping regions. These visualizations demonstrate that TransTCNet learns physiologically meaningful representations, in which the physiological properties of sEMG patterns are clustered as they occur in neural networks, rather than in arbitrary feature pairings.

 doi:10.20944/preprints202604.1498.v1

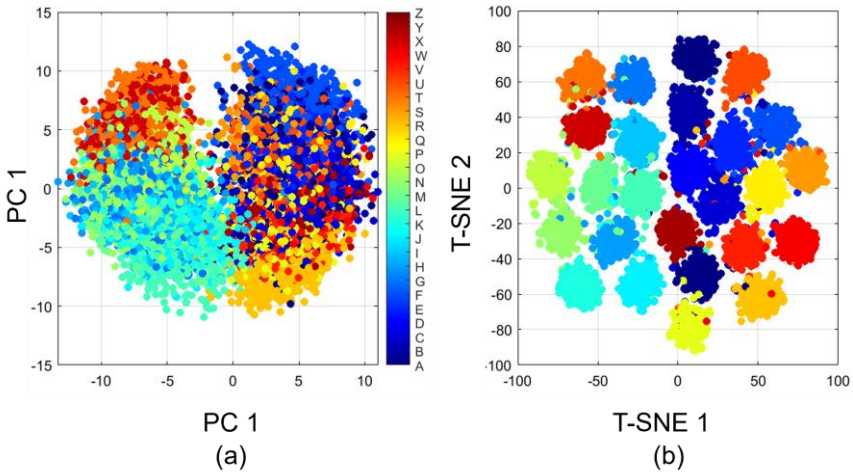


Figure 8. Feature space visualization via dimensionality reduction. (a) PCA projection of learned sEMG embeddings. (b) t-SNE transformation of feature representations for 26 alphabetical classes.

4.6. ROC Curve Analysis

Figure 9 shows the ROC curves for the 26 alphabet classes, indicating excellent discriminative performance throughout the classification task. The area under the curve (AUC) values for all classes are above 0.994. All classes achieve Area Under Curve (AUC) values exceeding 0.994, with classes L, P, Q, and X achieving the highest AUC of 0.998. near-perfect separability between positive and negative instances of each character. The low-density ROC curves in the upper-left region of the plot indicate that there is no trade-off between true and false positive rates across all classes. This consistent performance demonstrates a steady classification strength rather than occasional distinction confined to certain characters. 15 of the 26 classes demonstrated AUC values greater than 0.997, providing strong evidence that TransTCNet can achieve high sensitivity and specificity concurrently. However, Eleven classes showed less robust sensitivity and specificity. For all user types, it is important to maintain high sensitivity and specificity for both keystroke recognition and error detection, as degraded quality due to unrecognized or incorrect entries will negatively impact the user experience.

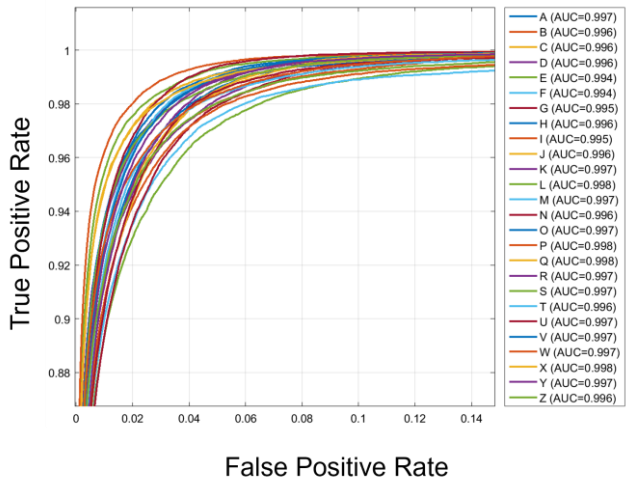


Figure 9. Receiver Operating Characteristic (ROC) curves for all 26 alphabetical classes, with Area Under Curve (AUC) values displayed for each character.

4.7. Prediction Confidence and Calibration Analysis

 doi:10.20944/preprints202604.1498.v1

4.7.1. Confidence Distribution Analysis

Figure 10 shows the distribution of prediction confidence for correct and incorrect classifications. Accurate predictions are highly right-skewed, with confidence values centered above 0.8. Conversely, the incorrect predictions are evenly distributed at the lower end of the confidence spectrum. The mean confidence for correct predictions is 0.9441, compared with 0.5716 for incorrect predictions, indicating a distinct 0.37-point difference between the two groups. The difference enables accurate estimation of uncertainty for low-confidence predictions (below 0.5) without false classifications, except for 32.11% accuracy. On the other hand, high-confidence predictions (>0.9) achieve 97.86% accuracy, indicating that confidence values are valid predictors of forecast reliability. The steep slope of the confidence distribution between correct and incorrect classifications provides a valuable tool for performing confidence-based error noting or assessment in real-time typing interfaces.

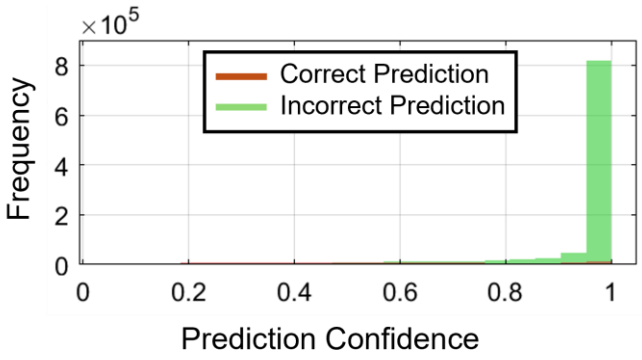


Figure 10. Distribution of prediction confidence values for correct versus incorrect classifications.

4.7.2. Model Calibration Assessment

As depicted in Figure 11, the model's predicted confidence relative to its actual accuracy shows high calibration across the 10 confidence levels. Especially as confidence grows from 0.4 and above, the calibration curve is very similar to the ideal diagonal. There is the greatest degree of variance at the low end of the confidence levels, specifically at the lowest level (0.0-0.1), where the predicted confidence falls slightly below its actual performance value. The sample sizes per bin range from 43 in the lowest bin to 826,441 in the highest; as a result, most predictions fall into the higher confidence ranges. There is an increase in accuracy from 32.11% in the 0.0-0.1 bin to 97.86% in the 0.9-1.0 bin, indicating that the confidence values represent a strong probability estimate of successful typing rather than simply overconfidence. This calibration information supports the use of the model TransTCNet for practical purposes, as it may be deployed in cases where there are confidence threshold requirements that call for user confirmation (i.e., typing error) and will help reduce typing error rates while not adversely affecting typing fluency.

 doi:10.20944/preprints202604.1498.v1

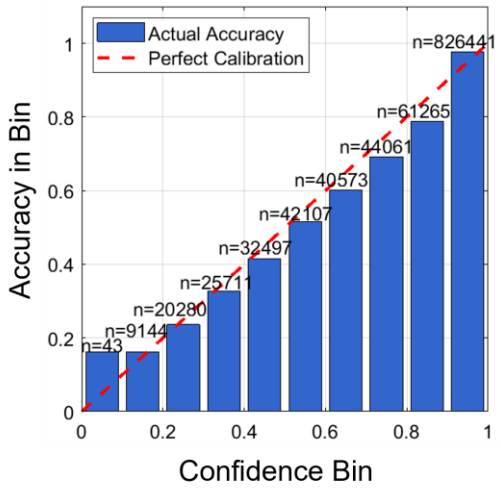


Figure 11. Calibration curve comparing predicted confidence to actual accuracy across ten confidence bins, with sample counts (n) indicated for each bin.

4.8. Error Pattern Analysis

Figure 12 identifies the ten most confusable character pairs in the classification task. The most frequently misclassified character pairs are E→D, J→N, and B→G. These error patterns illustrate the types of errors that result from the biomechanical similarities of these key presses, i.e., they all involve the use of the adjacent fingers or involve the use of similar muscles in the forearm. The presence of two pairs of bidirectional confusions (L↔O and U↔Y) indicates that all characters involved have similar muscle activity patterns, resulting in equal likelihood of confusion. The anatomical proximity of the most prevalent errors indicates that the spatial relationships among the fingers create a significant barrier to accurately classifying keys. For example, the left middle finger will be responsible for typing E and D, while the right index finger will be responsible for typing J and N, respectively. The identified confusion patterns can also be used to develop strategies to improve classification accuracy. Future development efforts may focus on refining the ability to discriminate between these character pairs through data augmentation strategies, attention to differences in temporal activation, or consideration of the spatial locations of the characters when typed. The relative number of errors in overall classifications (less than 2000 overall error rates across all pairs, compared with the overall number of predictions) highlights the model's effective performance, even in the presence of systematic error patterns.

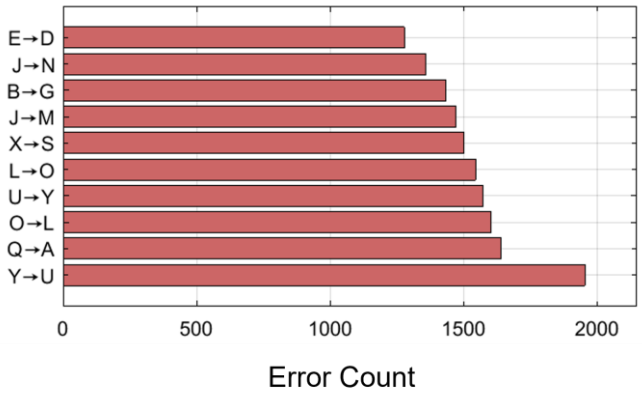


Figure 12. Top ten most frequently confused character pairs, with error counts indicated for each misclassification pattern.

4.9. Ablation Study

 doi:10.20944/preprints202604.1498.v1

A baseline model was constructed to assess the effect of each architectural component by sequentially adding the modules shown in Table 2. This study examines the contributions of temporal feature extraction (via dilated causal convolutions) and global context modeling (via transformer attention) to classification performance.

Table 2. Validation Accuracy for Model Variants in the Ablation Study, highlighting the Impact of each Architectural Component.

Model Variant	Architectural Components	Validation Accuracy (%)
Baseline	Standard 1D convolutional layers	48.66
+ Temporal Module	Dilated causal convolutions (dilation=2,4)	72.67
+ Global Context	Multi-head self-attention encoder	88.39
TransTCNet (Full)	Both temporal + global components integrated	96.53

4.9.1. Baseline: 1D-CNN

The baseline used standard 1D convolutional layers and achieved a validation accuracy of 48.66%. This showed that it could recognize basic spatial patterns but not temporal ones very well.

4.9.2. + Temporal Module (Dilated Causal Convolutions)

Adding dilated causal convolutions (dilation factors of 2 and 4) increased the accuracy to 72.67% (+24.01%). This increase in the size of the temporal receptive field enabled the capture of multiscale patterns of muscle activation that were critical for character discrimination.

4.9.3. + Global Context (Transformer Encoder)

The addition of a transformer encoder with multi-head self-attention increased accuracy by 88.39% (+15.72%). The attention model provided long-range dependencies and a dynamically weighted, informative temporal function.

4.9.4. TransTCNet: Full Architecture

The whole architecture, including both the temporal and global context modules and residual connections, achieved an accuracy of 96.53% (an improvement of 8.14% over the transformer-only approach). Hierarchical processing, comprising local temporal features followed by global sequence understanding, can yield synergistic effects that neither component alone can capture.

4.9.5. Architectural Justification for Fine-Grained Discrimination

The robust performance of TransTCNet in character-level sEMG typing recognition can be explained by its hierarchical architecture, which addresses the major challenges of fine-grained EMG classification. Dilated causal convolutions are used to capture multiscale temporal patterns necessary to distinguish muscle-activation dynamics. Simultaneously, the transformer encoder provides global context across the entire signal window. Such local-to-global processing enables discrimination between biomechanically similar keypresses that share the same spatial patterns but differ in their temporal structure. The resulting synergistic behavior and stable gradient flow achieved by the integrated design with residual connections provide stability in gradient flow and enable feature reuse, making the combined architecture perform better than either component alone. The highest validation accuracy of 96.53% and the average cross-participant validity and performance of 87.57% support the architecture's effectiveness in generalizing to individuals with diverse physiological attributes.

4.10. Comparison Models

 doi:10.20944/preprints202604.1498.v1

Table 3 compares the proposed TransTCNet architecture with the baseline models published with the original dataset. Previous methods, such as support vector machines (SVMs) employing custom statistical features and federated multi-layer perceptrons (MLPs), achieved accuracies ranging from 53.3% to 90.2% on the same 26-class character-level sEMG typing task. TransTCNet achieved a much higher mean accuracy of 95.03, demonstrating the importance of dilated causal convolutions, channel-wise recalibration, and multi-head self-attention within a single architecture. This architecture can effectively replicate both short-range motor activations and long-range contextual influences, which are essential for differentiating visually and kinaesthetically similar keypresses.

The present comparison is confined to conventional and federated learning methodologies. It is not comparable to the commonly employed paradigm of 2D-spectrogram-based CNNs for myoelectric interfaces. The methods convert sEMG signals into time-frequency representations, such as STFT or wavelet spectra. These representations are then further analyzed using convolutional networks to identify spatial-temporal patterns. Spectrogram-based CNNs help capture frequency-domain features, but they complicate preprocessing. They might obscure small-scale transient dynamics, which are critical for rapid decoding in keypress tasks. On the other hand, TransTCNet can process raw 1D sEMG sequences directly without spectral preprocessing. This facilitates rapid inferences, which is beneficial for real-time use.

Additionally, they were not compared with recurrent architectures such as LSTMs, BiLSTMs, or standard Transformers because no publicly available implementations were optimized for this dataset, and doing so would have made the evaluation pipeline less reliable. The upcoming research will focus on benchmarking and comparing deep learning baselines with those presented in this paper, including 1D and 2D paradigms, to assess trade-offs in accuracy, latency, generalization, and interpretability in real-world applications.

Table 3. Comparison of Classification Performance with Previously Published Methods on the Same SEMG Typing Dataset.

Model	Accuracy (%)	Comments
SVM + Handcrafted Features	87.4 ± 2.5	Baseline model using RMS, LOGVAR, WL, WAMP, ZC, AR1, AR2
SVM (Excl. spacebar class)	90.2 ± 2.1	26-class classification (A–Z only)
MLP (FedAvg)	53.3 ± 0.92	Shared model, no personalization
MLP (FedPer)	66.58 ± 1.01	Personalized classifier layers
MLP (pFedGP)	74.49 ± 0.72	Gaussian process with personalized heads
TransTCNet (Current model)	94.72% ± 0.31	Short-range temporal dependencies and long-range contextual patterns

4.11. Statistical Analysis

To analyze the accuracy and reliability of TransTCNet, one of the proposed systems to improve transfers, three types of transfer test analyses were used, in addition to an overall validation test of the system. The results of all three studies yielded an overall validation accuracy of 96.53 (σ=0.14); thus, all studies produced consistent, stable, and verifiable accuracy levels. The final comparison study between the original baseline 1D-CNN model (48.66%) and the study produced an overall average two-tailed p-value of <0.001, thereby indicating a statistically significant difference between the two baseline models, as due to pure chance variation. By conducting a cross-participant analysis across 19 participants evaluated, the study produced evidence of inter-participant generalization, as reflected in an average validation precision of 87.57 (σ=0.13) across all participants, with individual participants producing similar precision values and performance scores ranging from 87.30 to 87.81. The single classification ANOVA analysis demonstrated no statistically significant differences in model performance across participants ($F(18, 57) = 0.898$, $p = 0.587$), suggesting that the same model

 doi:10.20944/preprints202604.1498.v1

can accurately represent human physiology regardless of individual differences at the time of measurement. Consequently, the statistical results of this study provide sufficient support for the application and validation of TransTCNet and its intended use across various real-life situations and settings, demonstrating its generic applicability across user demographics and geographic locations.

5. Limitations and Future Work

The proposed TransTCNet model successfully decoded character-level typing motions at the keyboard by leveraging localized and global temporal patterns in raw sEMG signals. Multiple experiments validated its consistency, yielding a peak validation accuracy of 96.53% and a cross-participant mean accuracy of 87.57%. t-SNE visualizations of the learned embeddings showed clear separation into distinct clusters per alphabetical category of all 26 possible keypress categories. Computationally, TransTCNet is relatively efficient, with approximately 100 training iterations taking 1.51 hours on an NVIDIA RTX 5080, which is substantially faster than the initial models. The model is practical in terms of memory requirements for edge deployment. Despite these advantages, several limitations should be identified. To start with, although the 80:20 stratified split has shown within-session results, cross-subject analysis indicates an 8.96% difference (96.53 vs. 87.57), suggesting an effect of inter-subject variability. To address this performance gap, future work should employ Leave-One-Subject-Out (LOSO) cross-validation to assess the impact of the work on user-independent generalization and to inform the design of personalization strategies. Second, the data comprises only temporally discontinuous keypress trials, excluding idle/rest-state data, thereby limiting the model to single-gesture classification. The inclusion of continuous sEMG streams with a dynamic idle detector would make it more practical for real-world applications, such as typing interfaces. Third, comparing the most frequently mixed pairs of characters (E→D, J→N, B→G) indicates that biomechanical similarity remains a problem. Future versions could consider multitask learning methods that jointly optimize character classification and finger position estimation to improve discrimination between near keypresses. Fourth, although a fixed duration of 0.2 seconds offers the best trade-offs for standard typing speed, adaptive windowing approaches that account for neuromuscular timing differences may improve cross-subject robustness, particularly when subjects exhibit atypical motor development. Future perspectives must focus on some key points: (1) domain adaptation methods to decrease the cross-participant performance gap; (2) multimodal sensor fusion including inertial measurement units (IMUs) or keystroke dynamics to complementary motion information; (3), lightweight deployment optimizations, such as pruning, quantization, or knowledge distillation, to ensure edge device compatibility; and (4) longitudinal studies evaluating the level of performance maintenance under heavy usage and under different muscle fatigue conditions.

6. Conclusion

This study presents TransTCNet, a time- and context-sensitive deep learning model for detecting alphabetic keypresses via sEMG. The model efficiently acquires local and long-range signal relationships by combining dilated causal convolutions to extract multiscale temporal features and transformer-based attention mechanisms to model global context. In intra-session analysis, TransTCNet achieved the highest accuracy of 96.53%, whereas in cross-subject analysis, it had a mean accuracy of 87.57% across 19 subjects. Detailed evaluation shows good performance, with a mean AUC of 0.9965 across all 26 classes; high-confidence predictions achieve 97.86% accuracy; and a narrow range of performance variation among participants (0.51%). The representations of features, as in the model, exhibit physiologically important structure and dimensionality reduction; visualizations depict specific groupings based on character type; and maintain the connection between biomechanically related keys. Error analysis identifies patterns of systematic confusion that inform explicit improvements in subsequent cycles. TransTCNet offers an acceptable alternative to text entry via muscle movements, wearable systems, and real-time human-computer interaction, as it is highly accurate, computationally efficient, and highly generalizable. Its applications include

 doi:10.20944/preprints202604.1498.v1

prosthetic control, neuromuscular disability, augmentative communication, immersive AR/VR interfaces, and real-time muscular monitoring in sports and rehabilitation. The hierarchical architecture, that is, a combination of local temporal processing and global context modeling, is a principled approach to addressing the underlying problems in fine-grained sEMG classification, thereby improving the state of neural interfaces for silent, hands-free communication.

Author Contributions: Asif Ullah: Conceptualization; software; validation; investigation; data curation; visualization; writing—original draft. Zhendong Song: Supervision; writing—review & editing; funding. Waqar Riaz: Visualization, validation. Yizhi Shao: Supervision, data curation. Xiaozhi Qi: Funding Acquisition; Resources.

Funding: This work was supported in part by the Postdoctoral Foundation (Grant No: 6024331025K) and Research Projects of Shenzhen Polytechnic University (Grants: 6023310034K, and 6023310026K).

Declaration of conflict of interest: The author(s) declared no potential conflicts of interest concerning this article's research, authorship, and/or publication.

References

1. A. Yeo, B. W. Kwok, A. Joshna, K. Chen, and J. S. J. E. Lee, "Entering the next dimension: A review of 3d user interfaces for virtual reality," *Electronics,* vol. 13, no. 3, p. 600, 2024.
2. M. Zheng, M. S. Crouch, and M. S. J. I. S. J. Eggleston, "Surface electromyography as a natural human–machine interface: a review," *IEEE Sensors Journal,* vol. 22, no. 10, pp. 9198-9214, 2022.
3. T. Zaim, S. Abdel-Hadi, R. Mahmoud, A. Khandakar, S. M. Rakhtala, and M. E. J. B. Chowdhury, "Machine Learning-and Deep Learning-Based Myoelectric Control System for Upper Limb Rehabilitation Utilizing EEG and EMG Signals: A Systematic Review," *Bioengineering,* vol. 12, no. 2, p. 144, 2025.
4. E. Eddy, E. Campbell, S. Bateman, E. J. F. i. B. Scheme, and Biotechnology, "Big data in myoelectric control: large multi-user models enable robust zero-shot EMG-based discrete gesture recognition," *Frontiers in Bioengineering and Biotechnology,* vol. 12, p. 1463377, 2024.
5. I. Kyranou, K. Szymaniak, and K. J. S. D. Nazarpour, "EMG dataset for gesture recognition with arm translation," *Scientific Data,* vol. 12, no. 1, p. 100, 2025.
6. S. Zhang, H. Zhou, R. Tchantchane, and G. J. I. A. T. o. M. Alici, "A wearable human–machine-interface (HMI) system based on colocated EMG-pFMG sensing for hand gesture recognition," *IEEE/ASME Transactions on Mechatronics,* 2024.
7. C. Ma, C. Wang, D. Zhu, M. Chen, M. Zhang, and J. J. J. o. P. R. He, "The Investigation of the Relationship Between Individual Pain Perception, Brain Electrical Activity, and Facial Expression Based on Combined EEG and Facial EMG Analysis," *Journal of Pain Research,* pp. 21-32, 2025.
8. L. Adamov et al., "Comparative analysis of electrical signals in facial expression muscles," *BioMedical Engineering,* vol. 24, no. 1, p. 17, 2025.
9. A. Ullah et al., "Surface Electromyography-Based Recognition of Electronic Taste Sensations," *BioSensors,* vol. 14, no. 8, p. 396, 2024.
10. A. Salkanovic, D. Sušanj, L. Batistić, and S. J. S. Ljubic, "Beyond Signatures: Leveraging Sensor Fusion for Contextual Handwriting Recognition," *Sensors* vol. 25, no. 7, p. 2290, 2025.
11. A. Tigrini et al., "Intelligent human–computer interaction: combined wrist and forearm myoelectric signals for handwriting recognition," *Bioengineering,* vol. 11, no. 5, p. 458, 2024.
12. J. Eby, M. Beutel, D. Koivisto, I. Achituve, E. Fetaya, and J. Zariffa, "Electromyographic typing gesture classification dataset for neurotechnological human-machine interfaces," *Scientific Data,* vol. 12, no. 1, p. 440, 2025/03/15 2025, doi: 10.1038/s41597-025-04763-w.
13. N. A. Choudhury and B. Soni, "Enhanced complex human activity recognition system: A proficient deep learning framework exploiting physiological sensors and feature learning," *IEEE Sensors Letters,* vol. 7, no. 11, pp. 1-4, 2023.
14. E. Essa and I. R. Abdelmaksoud, "Temporal-channel convolution with self-attention network for human activity recognition using wearable sensors," *Knowledge-Based Systems,* vol. 278, p. 110867, 2023.

 doi:10.20944/preprints202604.1498.v1

15. S. Singh, N. A. Choudhury, and B. Soni, "Gait recognition using activities of daily livings and ensemble learning models," in *International Conference on Advances in IoT and Security with AI, 2023*, 2023: Springer, pp. 195-206.
16. S. Zhang et al., "Deep learning in human activity recognition with wearable sensors: A review on advances," *Sensors,* vol. 22, no. 4, p. 1476, 2022.
17. F. Demrozi, G. Pravadelli, A. Bihorac, and P. Rashidi, "Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey," *IEEE access,* vol. 8, pp. 210816-210836, 2020.
18. N. A. Choudhury and B. Soni, "An efficient CNN-LSTM approach for smartphone sensor-based human activity recognition system," in *2022 5th International conference on computational intelligence and networks (CINE)*, 2022: IEEE, pp. 01-06.
19. N. A. S. Putro, C. Avian, S. W. Prakosa, M. I. Mahali, J.-S. J. B. S. P. Leu, and Control, "Estimating finger joint angles by surface EMG signal using feature extraction and transformer-based deep learning model," *Biomedical Signal Processing and Control,* vol. 87, p. 105447, 2024.
20. A. Ullah, Z. Song, W. Riaz, X. Qi, and M. M. Hossain, "Hand Gesture-Based Biometric Verification and Identification Using Embedded-STQNet Deep Neural Architecture in Security-Oriented Systems," *IEEE Internet of Things Journal,* 2026.
21. M. A. A. Al-Qaness, A. Dahou, M. Abd Elaziz, and A. M. Helmi, "Multi-ResAtt: Multilevel residual network with attention for human activity recognition using wearable sensors," *IEEE Transactions on Industrial Informatics,* vol. 19, no. 1, pp. 144-152, 2022.
22. N. Hnoohom, A. Jitpattanakul, I. You, and S. Mekruksavanich, "Deep learning approach for complex activity recognition using heterogeneous sensors from wearable device," in *2021 Research, Invention, and Innovation Congress: Innovation Electricals and Electronics (RI2C)*, 2021: IEEE, pp. 60-65.
23. N. A. Choudhury, S. Moulik, and D. S. J. I. S. J. Roy, "Physique-based human activity recognition using ensemble learning and smartphone sensors," *IEEE Sensors Journal,* vol. 21, no. 15, pp. 16852-16860, 2021.
24. N. A. Choudhury and B. J. I. S. J. Soni, "An efficient and lightweight deep learning model for human activity recognition on raw sensor data in uncontrolled environment," *IEEE Sensors Journal,* vol. 23, no. 20, pp. 25579-25586, 2023.
25. A. Pradhan, "Electromyography-based Biometrics for Secure and Robust Personal Identification and Authentication," Doctoral dissertation, University of Waterloo, 2024.
26. A. Ullah, Z. Song, W. Riaz, and Y. Wang, "GTMH-TasteNet: Advanced Deep Learning for sEMG-Based Taste Sensation Recognition," *Tsinghua Science and Technology,* 2025.
27. M. Pourmokhtari and B. J. P. o. t. I. o. M. E. Beigzadeh, Part H: Journal of Engineering in Medicine, "Simple recognition of hand gestures using single-channel EMG signals," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine,* vol. 238, no. 3, pp. 372-380, 2024.
28. G. Liu et al., "Kinetic and Kinematic Sensors-free Approach for Estimation of Continuous Force and Gesture in sEMG Prosthetic Hands," *arXiv preprint arXiv,* p. 2407.00014, 2024.
29. M. Taheri, H. J. B. S. P. Omranpour, and Control, "Breast cancer prediction by ensemble meta-feature space generator based on deep neural network," *Biomedical Signal Processing and Control,* vol. 87, p. 105382, 2024.
30. S. J. Wetzel, S. Ha, R. Iten, M. Klopotek, and Z. J. a. p. a. Liu, "Interpretable machine learning in physics: A review," *arXiv preprint arXiv,* 2025.