Predicting Keystrokes from Electromyography Signals

Peike Li
Department of Biostatistics
University of California, Los Angeles
Los Angeles, CA 90095
peike239@g.ucla.edu

Wenxuan Karen Li
Department of Computational Medicine
University of California, Los Angeles
Los Angeles, CA 90095
wxli1028@g.ucla.edu

Debajyoti Chakrabarti
Department of Mechanical and Aerospace Engineering
University of California, Los Angeles
Los Angeles, CA 90095
debjyoti@g.ucla.edu

Qizhao Chen
Department of Mechanical and Aerospace Engineering
University of California, Los Angeles
Los Angeles, CA 90095
qizhaoc@g.ucla.edu

Abstract

This report explores the use of several neural network architectures to infer keyboard inputs from electromyography (EMG) signals, using data collected from subject 89335547 in the *emg2qwerty* dataset. In addition to the baseline CNN model, we experiment with alternative sequence modeling approaches, including bidirectional LSTM networks [4] and Transformer-based [5] layers implemented in PyTorch. These models are organized into different architectural configurations guided by both empirical testing and architectural intuition. After evaluating their performance, the most effective model is selected and further analyzed under varying electrode configurations to assess how well it performs when the number of electrode channels per band is limited to sixteen or fewer. Project link: **https://github.com/practice-lab-ucla/c247a**

1 Introduction

Recent advances in wearable sensing have enabled new human–computer interaction methods using physiological signals such as electromyography (EMG), where forearm sensors capture muscle activity to infer keystrokes without a physical keyboard. However, EMG signals are high-dimensional, noisy, and strongly time-dependent, requiring deep learning models that can effectively learn both spatial patterns across electrode channels and temporal dependencies in the signal.

In this work, we study several neural network architectures for EMG-based keystroke decoding using the *emg2qwerty* dataset [1]. We start with a convolutional neural network (CNN) baseline to capture spatial features from multi-channel EMG signals and further explore architectures designed to model temporal dependencies, including Long Short-Term Memory (LSTM) networks and Transformer-based layers. All models are implemented in PyTorch and evaluated under consistent training settings to compare their effectiveness in learning representations from EMG data.

Beyond model architecture, we also examine factors that influence decoding performance, including data augmentation strategies, the amount of available training data, and the effect of reducing the number of active electrode channels. These experiments help assess the robustness of the models under practical constraints such as limited data and lower sensor density, providing insights for the design of EMG-based typing systems.

2 Methods

In this project, we evaluate two neural network architectures for decoding keystrokes from the *emg2qwerty* dataset: a recurrent neural network (RNN) based model and a Transformer-based model. The RNN architecture uses LSTM units to capture temporal dependencies in sequential EMG signals, while the Transformer model leverages self-attention mechanisms to model long-range relationships across the input sequence. Both models are implemented in PyTorch and trained under the same experimental setting to enable fair comparison. Each model is trained for 150 epochs, and performance is evaluated based on their ability to accurately decode keystroke sequences from EMG recordings.

2.1 RNN Architecture (LSTM)

Because EMG signals are inherently sequential, we investigate an LSTM-based architecture to capture temporal dependencies in muscle activity during typing. To decode continuous keystrokes, we implement a recurrent neural network with LSTM units. The model takes windowed EMG feature vectors with input dimension 528, first processed by a multilayer perceptron (MLP) with hidden size 384 for feature extraction, followed by a three-layer LSTM encoder with hidden size 256 and dropout rate 0.3 for regularization. Training is performed using Connectionist Temporal Classification (CTC) loss [2] to align predicted character sequences with the EMG signals.

$$\text{MLP feature extractor} \rightarrow \text{3 LSTM layers} \rightarrow \text{Dropout} \rightarrow \text{CTC loss}$$

2.1.1 Data Augmentation

To improve model robustness to variability in EMG recordings, we incorporated several data augmentation techniques during training. First, low-level Gaussian noise was added to the EMG channels to simulate sensor noise. We also applied amplitude scaling and channel dropout prior to noise injection to introduce additional variability. Amplitude scaling multiplies the entire signal by a random factor (e.g., $0.8\times$ or $1.2\times$), simulating weaker or stronger keystrokes. Channel dropout randomly sets one complete electrode channel to zero with probability 0.5, encouraging the model to utilize information from multiple muscle signals rather than relying on a single dominant source. Gaussian noise is applied as the final augmentation step to ensure the artificial noise itself is not altered by previous transformations.

2.1.2 Training Data Scaling

To study the impact of training data availability on decoding performance, we conducted experiments with progressively smaller training sets. Specifically, we used the first $85\%, 75\%, 50\%,$ and 25% of the original training data to simulate scenarios in which a user completes fewer recording sessions.

2.1.3 Electrode Channel Reduction

To examine whether similar decoding performance could be achieved with a lower-density and more practical wearable design, we systematically reduced the number of active electrode channels. The baseline configuration uses 32 channels in total (16 per arm). Reduced-channel conditions were simulated during the forward pass by setting the signals from selected electrodes to zero. We evaluated two reduced-channel configurations: a 16-channel setting, in which every other electrode was deactivated (a 50% hardware reduction), and an 8-channel setting, where only one out of every four electrodes remained active (a 75% reduction), resulting in a substantially sparser measurement of the EMG signal.

2.1.4 Sampling Rate Analysis

To determine the optimal temporal sampling rate for processing the surface EMG (sEMG) signals, we conducted a temporal ablation study. The baseline dataset uses an effective sampling rate of 2000 Hz. We simulated lower sampling rates by downsampling the temporal dimension of the input sequence during the training and testing phases.

This was achieved by decimating the time dimension of the input tensors using a step size parameter, retaining every 2nd, 3rd, or 4th frame. This procedure yielded effective sampling rates of 1000 Hz, 667 Hz, and 500 Hz, respectively. Sequence length parameters were proportionally scaled down to ensure the CTC loss function accurately mapped the compressed inputs to the target character sequences.

2.2 Transformer

In addition to recurrent architectures, we investigated Transformer-based models to capture long-range temporal dependencies in EMG sequences. Unlike LSTMs, which process signals sequentially through recurrent hidden states, Transformers rely on self-attention mechanisms that allow each time step to directly attend to every other time step in the sequence. This enables more flexible modeling of global temporal relationships and allows parallel computation during training.

2.2.1 Transformer-Only Model

In the Transformer-only architecture, windowed EMG feature vectors are first projected into a lower-dimensional embedding space using a linear layer. Because self-attention is permutation-invariant, positional encodings are added to preserve temporal ordering information [5]. The resulting embeddings are then processed by stacked Transformer encoder layers composed of multi-head self-attention and position-wise feedforward networks with residual connections and layer normalization.

The encoder outputs are mapped to character logits through a final linear projection layer. As in the LSTM architecture, Connectionist Temporal Classification (CTC) loss is used to align predicted character sequences with EMG inputs without requiring explicit frame-level annotations.

The overall architecture can be summarized as:

$$\text{Input EMG} \rightarrow \text{Positional encoding} \rightarrow \text{Transformer encoder} \rightarrow \text{Linear output} \rightarrow \text{CTC loss}$$

While this architecture provides strong global temporal modeling capability, it operates directly on high-resolution EMG sequences. Since self-attention scales quadratically with sequence length, this increases computational cost and may make the model more sensitive to high-frequency noise present in the raw signal.

2.2.2 TDSConv + Transformer Model

To improve efficiency and robustness, we implemented a hybrid architecture that integrates Temporal Depthwise Separable Convolution (TDSConv) blocks [3] with Transformer encoder layers. The TDSConv front-end extracts local temporal features and performs temporal downsampling before global modeling is applied. This convolutional preprocessing captures short-term muscle activation patterns while reducing sequence length, thereby lowering the computational burden of subsequent self-attention layers.

After convolutional processing, the compressed feature sequence is passed through Transformer encoder layers to model long-range dependencies across keystroke events. The final sequence representations are projected to character logits and trained using CTC loss, consistent with the other architectures.

The overall architecture can be summarized as:

$$\text{TDSConv blocks} \rightarrow \text{Transformer blocks} \rightarrow \text{Linear output} \rightarrow \text{CTC loss}$$

By combining convolutional inductive biases with global self-attention, the TDSConv + Transformer model balances local temporal feature extraction and long-range dependency modeling, making it better suited for high-frequency biosignals such as EMG. In addition, we evaluate this architecture

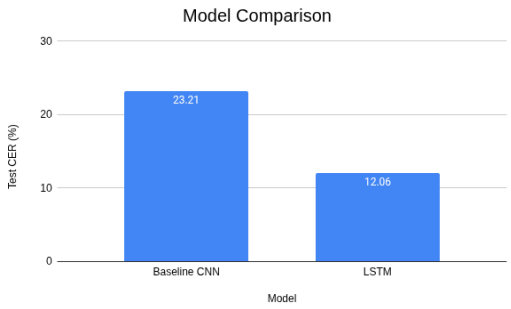
with temporal downsampling implemented through a 1D convolution layer applied along the time dimension. Specifically, we use a convolution with configurable kernel size, stride, and padding; setting stride reduces the temporal resolution while the kernel size and padding control the receptive field. This design allows controlled compression of the sequence length before the Transformer encoder, reducing computational cost while preserving contextual information.

3 Results

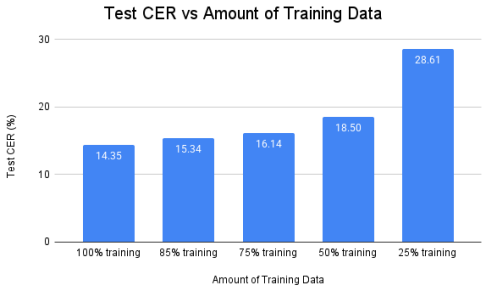
3.1 Baseline Model

As a benchmark, we first evaluate the baseline TDSConv model. After training for 150 epochs, the model achieved a training Character Error Rate (CER) of 2.39%, a validation CER of 19.39%, and a test CER of 23.21%. The low training error indicates that the model is able to fit the training data effectively, while the higher validation and test CER reflect a noticeable generalization gap. These results serve as a reference point for comparing the performance of the LSTM and Transformer-based architectures.

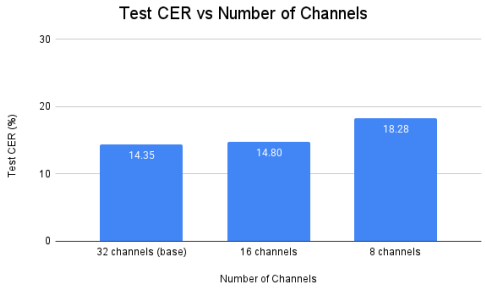
3.2 RNN Architecture (LSTM)



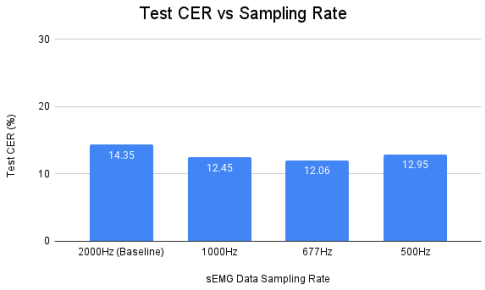
(a) Model Performance Comparison



(b) Test CER vs. Training Data Size



(c) Test CER vs. Number of Channels



(d) Test CER vs. Sampling Rate

Figure 1: Evaluation of the LSTM-based model under different experimental settings.

3.2.1 Model Comparison Across Baseline and RNN

We compare the test CER between the baseline CNN and the LSTM model (Fig. 1a). The CNN achieves a test CER of 23.21%, while the LSTM reduces it to 12.06%. This improvement suggests that the LSTM more effectively captures the temporal dependencies in EMG signals, leading to better decoding accuracy.

3.2.2 Data Augmentation

We also evaluate the impact of different data augmentation strategies on model performance. The baseline RNN achieves a test CER of 15.63%. Applying two augmentations—amplitude scaling and additive Gaussian noise—improves performance, reducing the test CER to 14.35%. In contrast,

adding channel dropout, which randomly zeros out one of the channels, results in a slightly higher test CER of 15.99%. These results suggest that moderate augmentation improves generalization, while additional channel dropout may remove useful signal information and slightly degrade performance.

3.2.3 Training Data Scaling

We study the impact of training data size on model performance. Fig. 1b shows that the test CER increases as the available training data decreases. Using the full dataset (100% training data) achieves the lowest test CER of 14.35%. When the training data is reduced to 85%, 75%, and 50%, the test CER gradually increases to 15.34%, 16.14%, and 18.50%, respectively. When only 25% of the training data is used, the error rate rises sharply to 28.61%. These results indicate that model performance strongly depends on the availability of sufficient training data, with significant degradation when the training set becomes too small.

3.2.4 Electrode Channel Reduction

We additionally analyze how the number of EMG channels affects model performance. Fig. 1c illustrates how the number of EMG channels affects model performance. Using all 32 channels (baseline) achieves the lowest test CER of 14.35%. Reducing the number of channels to 16 results in a slight increase in error to 14.80%, indicating that the model can still maintain similar performance with moderate channel reduction. However, further reducing the channels to 8 leads to a noticeable degradation in performance, with the test CER increasing to 18.28%. This trend suggests that while the model is somewhat robust to moderate reductions in input channels, a substantial decrease in available signal information significantly impacts decoding accuracy.

3.2.5 Sampling Rate Analysis

We finally evaluate how different EMG signal sampling rates affect model performance. As shown in Fig. 1d, the temporal ablation revealed a distinct U-shaped performance curve, demonstrating that a higher sampling rate does not strictly correlate with better transcription accuracy. The baseline 2000 Hz LSTM model exhibited the highest overall CER (14.35%). Reducing the sampling rate to 1000 Hz yielded a significant improvement, dropping the test CER to 12.45%. This suggests that the ultra-high frequency temporal data contains microscopic electrical noise that acts as an interference penalty rather than a useful feature for the recurrent neural network.

The optimal temporal resolution was found at 667 Hz, achieving the lowest test CER of 12.06% and the lowest specific rates for both substitutions (8.86%) and deletions (1.40%). This indicates a "sweet spot" where high-frequency static is effectively smoothed out, but the frame rate remains sufficiently high to capture the necessary kinematics of keystroke motions.

3.3 Transformer Architecture

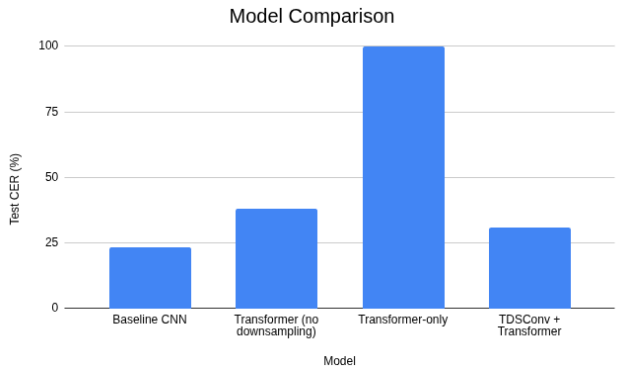


Figure 2: Model Performance Comparison

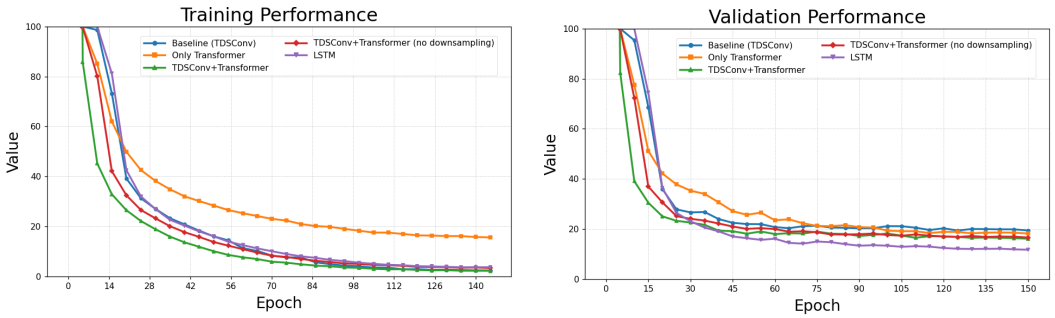
Figure 2 shows the training and validation CER curves for the Transformer-based architectures compared with the CNN and LSTM models. All models exhibit decreasing training CER over epochs.

However, their validation behaviors differ substantially. The LSTM achieves the lowest validation CER and demonstrates stable convergence. The CNN attains moderate validation performance with a relatively small train–validation gap. In contrast, the Transformer-only model shows a large discrepancy between training and validation CER, with validation error remaining high throughout training. The TDSConv + Transformer model performs better than the Transformer-only variant but still maintains higher validation CER than both the CNN and LSTM.

3.4 Model Comparison

Table 1: Test Character Error Rate (CER) for different model architectures.

Method	**Test CER (%)**
Baseline (CNN)	23.21
LSTM (RNN)	12.06
Transformer (no downsampling)	38.08
Transformer-only	100.00
TDSConv + Transformer	30.69



(a) Training Performance

(b) Validation Performance

Figure 3: Train / Validation performance plots for the experiment.

Table 1 reports the test CER for each evaluated architecture. The LSTM-based model achieved the lowest test CER at 12.06%, substantially outperforming all other models. The baseline CNN obtained a test CER of 23.21%. The Transformer variant without temporal downsampling resulted in 38.08%, while the Transformer-only model failed to converge effectively, yielding a CER of 100.00%. Among the Transformer-based approaches, the TDSConv + Transformer model achieved 30.69%, which remains notably higher than both the CNN and LSTM models.

Figure 3 further illustrates the training and validation CER curves for all architectures. Although all models reduce training error over time, their validation behaviors differ substantially. The LSTM demonstrates the most stable convergence and lowest validation CER, while the CNN achieves moderate generalization. In contrast, the Transformer-based models exhibit larger train–validation gaps, with the Transformer-only model showing clear failure to generalize. These trends align with the quantitative results shown in Table 1.

4 Discussion

All results in section 3.2 highlight the importance of modeling temporal dependencies for EMG-based text decoding. The RNN consistently outperforms the CNN baseline, suggesting that sequential architectures are better suited for capturing the temporal dynamics of keystroke-related EMG signals. Our ablation studies further indicate that moderate data augmentation and sufficient training data improve generalization, while excessive augmentation or substantial reductions in electrode channels can degrade performance. In addition, the sampling rate analysis suggests that an intermediate

temporal resolution balances signal fidelity and noise, leading to better decoding accuracy. Together, these findings emphasize that both temporal modeling and careful data preprocessing are critical for improving EMG-to-text transcription performance.

In contrast to the strong performance of the LSTM architecture, the Transformer-based models performed substantially worse across all configurations. The Transformer-only model failed to converge effectively, resulting in a test CER of 100.00%, while the TDSConv + Transformer model achieved 30.69%, still higher than the baseline CNN (23.21%) and the LSTM (12.06%).

We also observed a notable gap between validation and test performance for the Transformer models, with validation CER lower than test CER. This suggests limited generalization to unseen data and potential overfitting to the training and validation distributions. The high model capacity and flexibility of self-attention mechanisms may partially explain why the Transformer performs worse than the CNN baseline. Transformer architectures allow each time step to attend globally across the entire sequence, introducing substantially more parameters and weaker structural constraints. However, EMG signals are relatively noisy, high-frequency, and dominated by local temporal patterns associated with individual keystrokes.

In contrast, convolutional architectures impose strong local inductive bias through shared kernels and localized receptive fields, which naturally emphasize short-term signal structure while suppressing high-frequency noise. This inductive bias appears better aligned with the characteristics of EMG data. As a result, the Transformer may overfit spurious global correlations and generalize poorly compared to the more structured CNN baseline.

Overall, these results indicate that architectures with stronger sequential inductive bias, such as LSTMs, are better suited for EMG-based keystroke decoding under the current dataset and training regime.

4.1 Future Directions

Future work will focus on designing a more locally constrained Transformer architecture tailored to EMG signals. In particular, we plan to incorporate locality-aware self-attention mechanisms, such as restricted attention windows or hierarchical attention structures, to better capture short-term temporal dynamics while limiting the influence of noisy long-range correlations. By introducing stronger local inductive bias into the Transformer, we aim to improve generalization and bridge the performance gap observed between attention-based models and LSTMs.

Project link: **https://github.com/practice-lab-ucla/c247a**

References

[1] Jonas Frey et al. emg2qwerty: A large dataset with baselines for touch typing using surface electromyography. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[2] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 2006.

[3] Awni Hannun et al. Sequence-to-sequence speech recognition with time-depth separable convolutions. In *Proc. Interspeech*, 2019.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[5] Ashish Vaswani et al. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.