Evaluating LSTM, GRU-CNN, and Transformer for sEMG-to-QWERTY Decoding Using CER

Yuanyuan Xue
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095
yuanyuanxue@g.ucla.edu

Abstract

This project employs RNN architectures, including Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) layers, alongside the Transformer architecture, to predict characters typed from surface electromyography (sEMG) signals. Performance is evaluated using the character error rate (CER), where a lower CER indicates better accuracy. Comparing the three architectures against a baseline model, we find that the hybrid GRU-CNN model achieves the best test CER, followed by the LSTM model, the baseline, and the Transformer model. These results highlight the efficacy of hybrid recurrent-convolutional approaches for sEMG-based typing prediction, with the Transformer showing the weakest generalization.

1 Introduction

Surface electromyography (sEMG) signals offer a promising avenue for decoding human intent, such as typing, with applications in neural interfaces for communication and control. The emg2qwerty dataset(1) provides a rich resource to explore this task, pairing sEMG recordings from wristbands with ground-truth QWERTY keystrokes. While the baseline Temporal Depthwise Separable (TDS) model achieves reasonable performance, minimizing CER on a single subject's data remains critical for practical, user-specific deployment, where low error rates enhance usability and reliability.

This project investigates the question of how to minimize validation and test CER for a single subject (ID #89335547) by comparing advanced neural architectures against the baseline TDS model. High CERs in sEMG-to-keystroke decoding can stem from the complexity of temporal dependencies and inter-electrode variability in sEMG signals, which standard convolutional approaches like TDS may not fully capture. We hypothesize that architectures adept at modeling sequential data—such as recurrent neural networks (RNNs) and Transformers—could outperform the baseline by better leveraging temporal patterns and contextual relationships across the 32 sEMG channels. To test this, we evaluate three architectures: an LSTM, a classic RNN for temporal data(4); a hybrid GRU-CNN, which combines recurrent and convolutional strengths; and a Transformer, known for its attention-based sequence modeling. This exploration is motivated by the need for robust, personalized decoding in real-world settings, where minimizing CER directly impacts the effectiveness of sEMG-based typing interfaces.

Our approach builds on the baseline implementation from (2), extending it with these architectures to assess their efficacy on the single-user dataset. By focusing on a single subject, we aim to uncover architecture-specific insights that could inform user-tailored models, stepping towards broader generalization. The results offer a comparative analysis of how these post-CNN designs influence CER, providing a foundation for optimizing sEMG decoding performance in practical scenarios.

2 Methods

2.1 Architecture Pipeline

The pipeline begins with sEMG data preprocessing using `transforms.py`, applying data augmentation via random band rotations and computing log spectrograms with 33 bins over 0-1000 Hz (stride 16, 2 kHz $\rightarrow$ 125 Hz). Batch normalization is then applied on the log spectrograms for each of the 16 electrodes across 2 bands, yielding a feature set of 33 frequency bins per electrode. These processed signals, paired with ground-truth keystrokes, are input into a PyTorch Lightning framework (`lightning.py`) for training. Each model outputs frame-wise character probabilities, decoded using CTC loss to align variable-length sEMG sequences with keystroke labels. Training employs the Adam optimizer over 50 epochs, conducted on a single user's data.

2.2 LSTM-Based Model

The LSTM-based model leverages Long Short-Term Memory (LSTM) networks to capture temporal dependencies in sEMG signals, aiming to minimize CER by modeling keystroke sequences. It processes spectrograms flattened to $[8000, N, 384]$ (time, batch, features) after `MultiBandRotationInvariantMLP`. In our model, a batch size of $N = 64$ is used. A bidirectional LSTM with 4 layers (256 hidden units each, dropout 0.1) encodes the sequence, outputting $[8000, 64, 512]$ by leveraging past and future context. Then, a `TDSFullyConnectedBlock` applies two linear layers with ReLU and a residual connection, followed by a linear layer projecting back to 384 features, and layer normalization. The `LinearWarmupCosineAnnealingLR` scheduler is employed with 5 warm-up epochs and max learning rate 0.002. This deep, bidirectional design enhances temporal modeling over the baseline TDS, potentially reducing CER by capturing long-range sEMG patterns critical for single-subject decoding.

2.3 Hybrid GRU-CNN Model

The hybrid GRU-CNN combines Gated Recurrent Units (GRUs) with Convolutional Neural Networks (CNNs) to extract spatial and temporal features from sEMG signals. A `TDSConvEncoder` (3 blocks, 32 channels, kernel width 32) first processes inputs of shape $[8000, N, 528]$ from `MultiBandRotationInvariantMLP`, and `TimePool` downsamples time by factor of 4. Two 1D CNN layers (64 channels each, kernel size 5, padding 2) with ReLU and batch normalization reduce features to $[8000, N, 64]$, preserving sequence length. A 3-layer bidirectional GRU (512 hidden units, dropout 0.2) then models temporal dependencies, yielding $[8000, N, 1024]$, followed by a linear layer projecting back to 512 features. This model utilizes a `StepLR` scheduler with a learning rate of 0.005 and a batch size of $N = 64$. This design combines CNN efficiency with deep GRU sequence modeling, aiming to enhance CER by capturing both local and long-range sEMG patterns. A batch size of $N = 64$ is used during training.

2.4 Transformer-Based Model

The Transformer-based model employs attention mechanisms to capture global sEMG dependencies, with the aim of minimizing CER. A `TDSConvEncoder` (3 blocks, 24 channels, kernel width 32) is used first to refine features, followed by `TimePool` (kernel size 4, stride 4), reducing the sequence to $[2000, N, 768]$, which reduces costs for attention computation. A Transformer encoder (2 layers, 8 heads, 512-unit feed-forward, dropout 0.3) with positional encoding then models dependencies. The `LinearWarmupCosineAnnealingLR` scheduler is used with 5 warm-up epochs and a maximum learning rate of 0.001, with weight decay of 1e-4. A batch size of $N = 64$ is used. This design leverages attention for contextual modeling, potentially reducing CER by focusing on key sEMG patterns despite temporal downsampling.

3 Results

All models were trained for 50 epochs. Validation CERs were kept track of for each epoch, and final test CERs were determined.

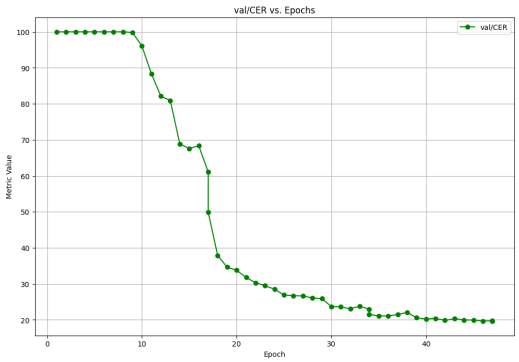

Figure 1: Validation CER for Baseline Model.

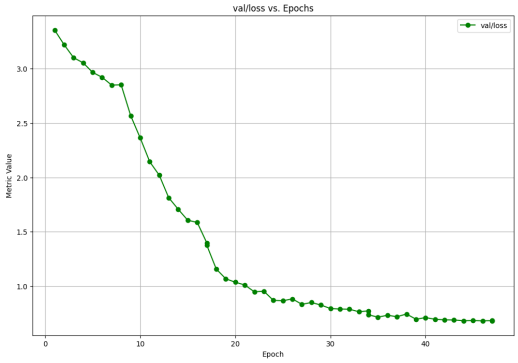

Figure 2: Validation Loss for Baseline Model.

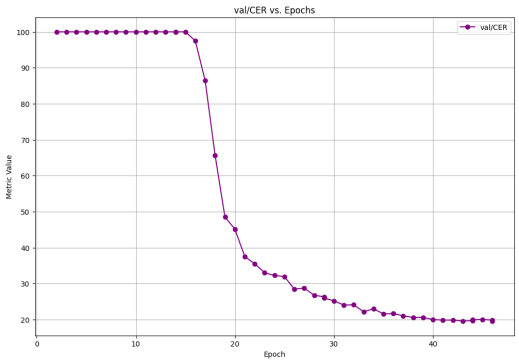

Figure 3: Validation CER for LSTM Model.

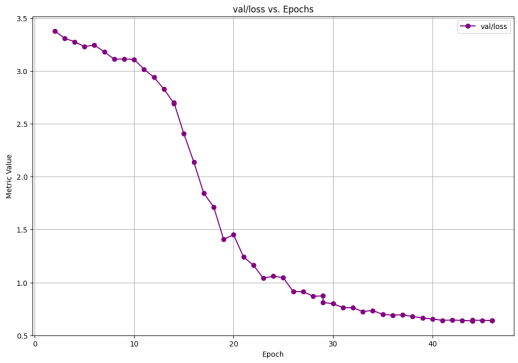

Figure 4: Validation Loss for LSTM Model.

3.1 Baseline Results

The baseline model achieved a validation CER of 19.74 and a test CER of 22.41, indicating reasonable generalization. Training dynamics revealed a validation CER of 100 for epochs 1–9, with loss decreasing steadily. The CER dropped most rapidly between epochs 10 and 20, then decreased slowly to around 20 at epoch 40 and thereafter. This suggests effective learning after initial stagnation.

3.2 LSTM-Based Model Results

The LSTM-based model achieved a validation CER of 19.61 and a test CER of 20.81, reflecting reasonable generalization. During training, the validation CER remained at 100 for epochs 1–16, before dropping sharply from 97.50 at epoch 17 to 37.53 at epoch 22. Subsequently, the CER decreased gradually, stabilizing around 20 by epoch 40.

3.3 Hybrid GRU-CNN Model Results

The hybrid GRU-CNN model achieved a validation CER of 16.44 and a test CER of 16.68, demonstrating strong generalization. It exhibited the fastest convergence among all models, reaching a validation CER of 19.94 by epoch 10. The CER then decreased gradually, stabilizing around 16.5 by epoch 20, aligning with the validation loss trend. Early stopping at epoch 35 was triggered due to this stabilization.

3.4 Transformer-Based Model Results

The Transformer-based model achieved a validation CER of 14.86 and a test CER of 40.80, indicating overfitting on the single-subject data. The validation CER remained near 100 for the first 5 epochs, then dropped earlier than the baseline and LSTM models but later than the hybrid GRU-CNN model, stabilizing around 15 by epoch 40.

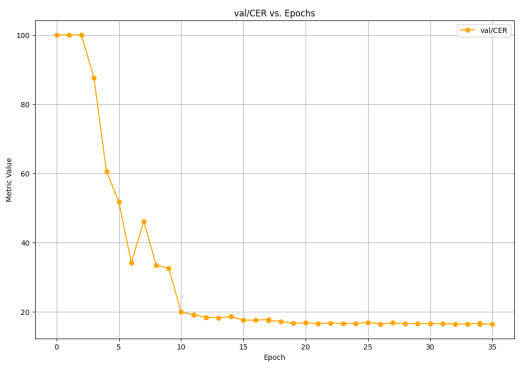

Figure 5: Validation CER for GRU-CNN Model.

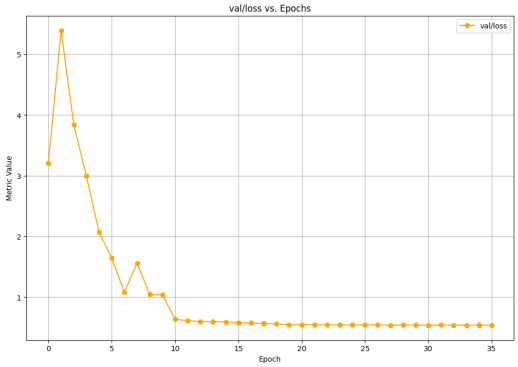

Figure 6: Validation Loss for GRU-CNN Model.

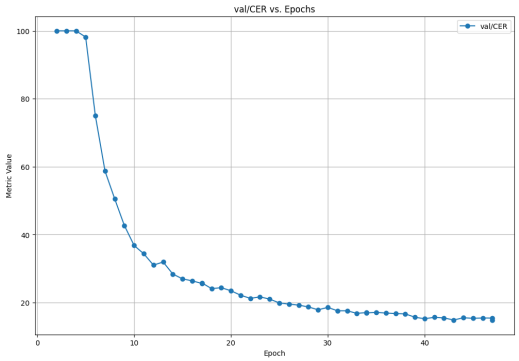

Figure 7: Validation CER for Transformer Model.

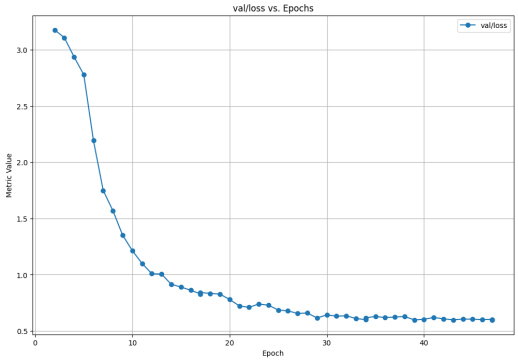

Figure 8: Validation Loss for Transformer Model.

4 Discussion

The Transformer, LSTM, and hybrid GRU-CNN models exhibit distinct performance profiles compared to the baseline (validation CER: 19.74, test CER: 22.41). The hybrid GRU-CNN outperforms all, with a validation CER of 16.44 and a test CER of 16.68, followed by the LSTM (validation CER: 19.61, test CER: 20.81) and the Transformer (validation CER: 14.86, test CER: 40.80). These results highlight varying degrees of generalization and efficiency in single-subject sEMG decoding, driven by each model's architectural characteristics.

The hybrid GRU-CNN's outstanding performance stems from its balanced design. Its CNN layers reduce dimensionality early (512 to 64 channels), enabling the 3-layer bidirectional GRU to efficiently model temporal sEMG patterns. This synergy results in rapid convergence (CER of 19.94 by epoch 10) and stability around 16.5, with a minimal generalization gap (0.24). We hypothesize that avoiding aggressive downsampling preserves critical keystroke-related sEMG signals, making the GRU-CNN particularly robust for this task.

In contrast, the Transformer achieves the lowest validation CER but overfits severely (test CER: 40.80). Its attention mechanism excels at capturing global dependencies, converging early (CER drop after epoch 5), but the `TimePool` layer's 4x temporal downsampling (8000 to 2000 frames) likely discards fine-grained patterns essential for generalization(3). This large gap (25.94) underscores the Transformer's sensitivity to temporal resolution in single-subject decoding.

The LSTM converges slower (CER drop after epoch 16) but generalizes better than the Transformer, with a smaller gap (1.2). Its 4-layer bidirectional structure (256 hidden units) processes full 8000-frame sequences without downsampling, aiding robustness, though its higher CER suggests limited feature extraction capacity. The baseline, stabilizing at a CER of 20 after a drop between epochs 10–20, relies on the TDSConvEncoder's temporal convolution but lacks the recurrent depth of LSTM/GRU, resulting in moderate performance.

These findings suggest that architectures that balance feature extraction and temporal modeling, such as GRU-CNN, potentially excel in single-subject sEMG decoding. Preserving temporal resolution could be crucial, as downsampling may has harmed generalization. Future work could explore multi-resolution attention for Transformers to retain fine-grained patterns, potentially improving their robustness while maintaining training efficiency.

References

[1] Viswanath Sivakumar, Jeffrey Seely, Alan Du, Sean R Bittner, Adam Berenzweig, Anuoluwapo Bolarinwa, Alexandre Gramfort, and Michael I Mandel. emg2qwerty: A large dataset with baselines for touch typing using surface electromyography, 2024.

[2] Joe Lin. emg2qwerty GitHub repository, 2024. `https://github.com/joe-lin-tech/emg2qwerty`.

[3] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Jinfeng Hu, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022. `https://arxiv.org/abs/2202.07125`.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. `https://doi.org/10.1162/neco.1997.9.8.1735`.