EMG2Keys: Optimizing Deep Learning Factors for Accurate sEMG-Based Keystroke Prediction

Kelsey Lin
Master of Engineering
UCLA
klin0522@g.ucla.edu

Muzhen Shen
Master of Engineering
UCLA
muzhen612@g.ucla.edu

Brayden Woods
Master of Science in Engineering Online
UCLA
braydenwoods@g.ucla.edu

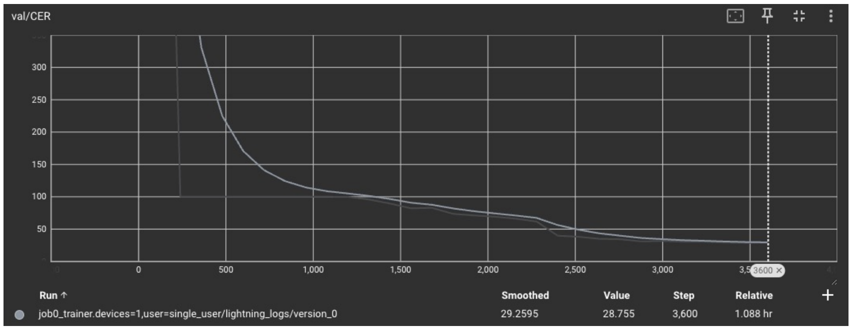
Abstract

In this project, we explore which factors influence the accuracy of deep-learning-based keystroke prediction from surface electromyography (sEMG) signals. We applied Meta's baseline Temporal Decoding System (TDS) model previously trained on a single subject within their emg2qwerty dataset, which contains synchronized sEMG signals and the ground-truth QWERTY keystrokes. Our primary focus was to understand how electrode channels, dataset size, sampling rate, preprocessing, and architectural choices impact the Character Error Rate (CER). In addition to improving the baseline, we sought to gain insights into these relationships by experimenting with various configurations. Our results shed light on key data and architectural factors affecting sEMG-based keystroke prediction and can aid in the future development of more efficient EMG-driven interfaces.

1 Introduction

Surface electromyography (sEMG) is a promising non-invasive technique that can measure motor unit action potentials, making it promising for use even in amputees and individuals with paralysis. The emg2qwerty dataset from Meta allows for the decoding of sEMG activity to predict the keystrokes a person is typing. The dataset provides 32-channel, preprocessed sEMG recordings from both wrists, along with synchronized ground-truth keystroke logs. A single subject from this dataset, ID #89335547, was used to train Meta's baseline TDS model and evaluated using CER as the performance metric. After running the baseline code, we achieved a benchmark CER of 29.

Rather than solely relying on improving the baseline TDS model, we also conducted this study to guide future work on sEMG-based technology. The research discussed by Merletti et al. [2021] reveals current limitations affecting the widespread adoption of sEMG in clinical settings. sEMG data are susceptible to noise and not easily interpretable without extensive expertise in the field. Additionally, the lack of large-scale clinical trials makes it difficult to validate and generalize sEMG-based systems. These challenges motivate our project, and they shed light on current setbacks in optimizing sEMG-based deep learning models on a larger scale.



Run ↑	Smoothed	Value	Step	Relative
job0_trainer.devices=1,user=single_user/lightning_logs/version_0	29.2595	28.755	3,600	1.088 hr

Figure 1: Validation CER for the baseline model

Our Github repository is forked from the original repo, but all code we built on top of the baseline is available in the main branch. This approach allows us to both preserve the original implementation while clearly distinguishing our contributions and improvements.

There remains a need to better understand the various factors that influence the accuracy of baseline predictions. To address this, we explored the effects of adjusting electrode channels, data size, sampling rate, data preprocessing, and architecture choice using the provided baseline model and dataset. These factors were independently explored to examine their individual impact on model performance, with the best-performing configurations improving the baseline TDS model and producing a lower CER.

2 Methods

2.1 Experimental Framework

To explore the factors that influence accurate keystroke prediction in a structured way, we divided our approach into three key areas: analysis of data characteristics versus decoding performance, data pre-processing, and architectural changes. Each was evaluated by altering the baseline and assessing its impact on CER. All models were trained using an NVIDIA T4 GPU environment within Google Colab for 30 epochs. Our final model combines the best architecture with the best data preprocessing techniques.

2.2 Data Characteristic Modifications

Channel Size Reduction The baseline TDS model utilizes all 32 channels (16 electrode channels per wrist). To examine the effect of channel reduction, we implemented dynamic channel selection in the baseline pipeline. We introduced `self.num_channels` as a configurable parameter, and modified `data.py` to support 8 channels instead of the original 16. The `train.py` file was altered to ensure correct handling of our new parameter, and `transforms.py` and `lightning.py` also reflected changes in channel dimension. Finally, we added '`num_channels: 8`' into the `tds_conv_ctc.yaml` under modules for easy channel size selection as well as halved '`in_features:`' to 264 from 528.

Training Data Size Variation To examine the effect of training data size on model performance, we incrementally decreased samples from 12, 8, and 4, each being a separate training instance on the baseline architecture. This exploration aimed to determine if there is a minimum viable data size that still yields effective keystroke predictions.

Sampling Rate Adjustments The EMG data, originally sampled at 2kHz (2000 Hz), was resampled using a custom ResampleEMG transform class. This implementation allowed for flexible adjustment of the sampling rate by selecting every n-th sample from the original data based on the desired target rate. When downsampling from 2kHz to lower rates such as 1kHz, the transform simply took every other sample, which reduced computational requirements while maintaining signal characteristics. During training, each epoch took about half the time to train as it did in the baseline, allowing for much quicker results. This approach maintained the structured array format of the EMG data with its electrode channels intact. Testing revealed that maintaining the original 2kHz sampling rate was critical for performance, as resampling to lower rates significantly degraded CER, with rates below 1.5kHz resulting in nearly complete decoding failure.

2.3 Signal Processing and Preprocessing

To improve generalization and robustness, we implemented two key preprocessing techniques:

- **EMG Noise Injection:** We augmented raw EMG signals with controlled Gaussian noise to simulate natural variations present in real-world usage scenarios. This technique was implemented as a custom transform that specifically targets EMG fields while preserving temporal alignment.
- **Electrode Dropout:** To simulate real-world scenarios where electrode contact may be intermittent or degraded, we implemented a channel dropout technique that randomly zeros out specific electrode channels.

We tested two configurations with varying levels of noise and dropout probability:

Parameter	Configuration 1	Configuration 2	Change
Noise Level	0.005 (0.5%)	0.01 (1%)	2× increase
Dropout Probability	0.1 (10%)	0.2 (20%)	2× increase
Electrodes per Band	1	1	No change

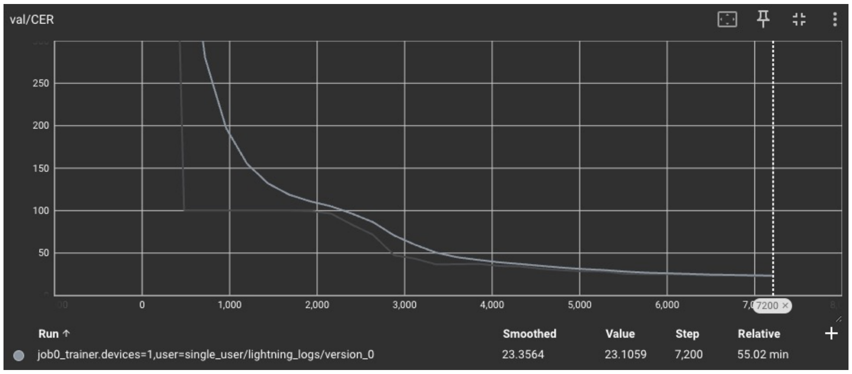
Table 1: Preprocessing parameters for the two configurations tested

Table 2: Training and Testing CER Results for Different Preprocessing Configurations

Configuration	Training CER (%)	Testing CER (%)
Configuration 1 (0.5% noise, 10% dropout)	26.67	27.71
Configuration 2 (1% noise, 20% dropout)	34.56	34.36

The parameters were integrated into the transformation pipeline before spectrogram conversion to ensure they affected the raw signal characteristics.

2.4 Architecture Modifications



Run ↑	Smoothed	Value	Step	Relative
job0_trainer.devices=1,user=single_user/lightning_logs/version_0	23.3564	23.1059	7,200	55.02 min

Figure 2: Validation CER for the hybrid TDS+GRU model

TDS+GRU Hybrid To explore the effect of different architectures on prediction accuracy, we implemented a hybrid model combining the foundational architecture of a TDS model with a Gated Reccurent Unit (GRU) encoder. This exploration aimed to understand whether the incorporation of recurrent elements into the predominantly convolutional baseline architecture could better capture the temporal dependencies inherent in keystroke sequences from sEMG signals. GRU-based architectures have shown improved classification performance by combining convolutional feature extraction with sequence learning, making this hybrid approach a strong candidate [Vijayvargiya et al., 2022]. In order to integrate GRU into the TDS baseline architecture, we added a TDSConvGRUEncoder class within `modules.py`.

TDS+Transformer Hybrid The second architecture we explored was a TDS+Transformer hybrid that replaced the GRU encoder with a transformer with self-attention mechanisms and positional embeddings. In order to integrate a transformer into the TDS baseline architecture, we added a `TDSConvTransformerEncoder` class within `modules.py` containing a default of 4 attention heads and 2 transformer layers with dropout at 0.1. A positional encoder class was also implemented using sinusoidal embeddings to maintain temporal order. The `window_length` was set to 4000 with padding at [900, 100] to provide sufficient but focused context for the attention heads. The TDS+Transformer model was trained with an AdamW optimizer to help improve stability.

3 Results

3.1 Data Characteristic Modifications

Effect of Channel Size Reduction After training the TDS baseline architecture on 16 total channels, half of the original 32 channels, our findings support our original hypothesis that lowering channels will result in less information for the model during training. The CER of the test increased to 43.96, a significant performance drop compared to the baseline CER of 29.

Effect of Training Data Size In our exploration of the effects of sample size on model performance, we observed a strong positive correlation between the number of training samples and the accuracy of the prediction. When reducing the training data from the full 16 samples to smaller subsets, we documented a clear performance degradation. The baseline CNN model achieved a Character Error Rate (CER) of 29 with all 16 samples, but performance declined substantially with each reduction in training data: 12 samples yielded a CER of 52.32, 8 samples resulted in a CER of 83.23, and with only 4 samples, the CER reached 100.

Sampling Rate Effects The original system operated at a 2000 Hz (2 kHz) sampling rate. Our attempts to reduce computational demands through downsampling to 1500 Hz or 1000 Hz resulted in severe performance degradation, with Character Error Rates (CER) reaching 96.

3.2 EMG Signal Preprocessing Performance

Table 1 demonstrated significant improvement over baseline, achieving a validation CER of 26.67% and test CER of 27.71% with only a 1.04% validation/test gap, indicating strong generalization capabilities. While Configuration 2 showed worse overall results (validation CER 34.56%, test CER 34.36%), the validation/test gap narrowed further to just 0.2%, suggesting increased augmentation reinforced generalization at the expense of accuracy.

Breaking down error types reveals specific patterns across configurations. The most dramatic change occurred in the Insertion Error Rate (IER), which increased by 6.62 percentage points in validation (70% relative increase) and 5.77 percentage points in testing (77% relative increase) when moving from table 1 to 2. This suggests stronger augmentation made the model oversensitive to signal variations, causing it to interpret noise as valid keystrokes. The Substitution Error Rate (SER) exhibited an interesting pattern during training, starting near zero and climbing to approximately 17% toward the end of training, particularly in Configuration 2 (Appendix B). This indicates that as training progressed with heavily augmented data, the model increasingly struggled to discriminate between similar characters.

The moderate augmentation in table 1 provided an optimal balance between regularization and signal fidelity through several mechanisms. It improved noise robustness by making the model less sensitive to natural variations in EMG signals. It enhanced fault tolerance by forcing the model to learn redundant representations across multiple electrodes. It also served as an effective regularizer, preventing overfitting to specific electrode patterns. In contrast, table 2's performance degradation resulted from signal-to-noise ratio reduction, spatial information loss as critical electrode channels were more frequently missing, and increased character confusion between similar muscle activation patterns.

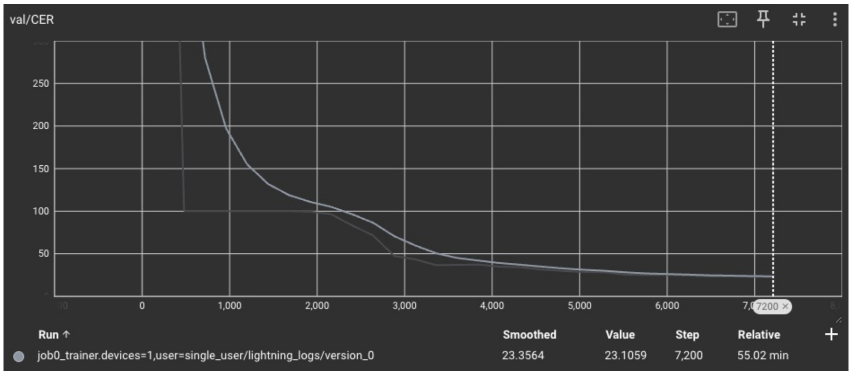
Substitution Error Rate Trends The isolated increase in SER, while other error metrics remained relatively stable (Appendix C), indicates that our preprocessing techniques specifically degraded the model's ability to distinguish between similar characters. Increasing noise from 0.005 to 0.01 appears to obscure subtle EMG signal differences that distinguish similar characters (e.g., 'a' vs 's' or 'm' vs 'n') due to their close proximity to each other on the keyboard. These characters likely produce similar muscle activation patterns that become increasingly indistinguishable when noise is amplified. Additionally, the increased dropout probability likely eliminated critical spatial information needed for character differentiation. When specific electrodes capturing unique motor unit activation patterns are dropped, the model struggles to distinguish similar keystrokes.

3.3 Architecture Performance Comparison

TDS+GRU The initial hybrid TDS+GRU model yielded a test CER of approximately 33.50. When expanding the information span by increasing the `window_length` from 8000 to 12000 samples and proportionally adjusting padding from [1800, 200] to [2700, 300], performance degraded to a test CER of 88.33.

Reducing the temporal context window to 4000 samples and adjusting the padding to [900, 100] achieved a performance improvement with a test CER of 22.91, representing a 21% reduction from baseline.

TDS+Transformer Our hybrid model TDS + transformer had a learning rate of 1e-3 using the AdamW optimizer with weight decay set at 5e-4. It was trained with 4 attention heads, 2 layers, and dropout of 0.1. Using `window_length` of 4000 and padding of [900, 100], it achieved a final test CER of 32.63.



Run ↑	Smoothed	Value	Step	Relative
job0_trainer.devices=1,user=single_user/lightning_logs/version_0	23.3564	23.1059	7,200	55.02 min

Figure 3: Validation CER for the best model

3.4 Best Model Performance

By combining a TDS+GRU architecture with optimized window size and strategic data augmentation, we achieved a validation CER of 22.57 and test CER of 22.87, representing a 21.1% improvement over the baseline CER of 29.

4 Discussion

4.1 Impact of Data Characteristic Modifications

Importance of Channel Count The significant performance drop observed when reducing channels highlights the importance of channel count, as removing channels directly limits the model's ability to learn robust representations of electrical activity from the muscles. This suggests that future work on sEMG-based decoding should explore how to optimize electrode placement or determine which channels typically contribute most to keystroke prediction.

Data Requirements for sEMG Prediction The pronounced performance drop with reduced training data illustrates the data-hungry nature of deep learning approaches for sEMG-based keystroke prediction and underscores the critical importance of sufficient training examples for establishing the complex relationships between muscle activation patterns and corresponding keystrokes. The near-linear deterioration suggests that future improvements might benefit significantly from expanded datasets, particularly when developing systems intended for real-world deployment.

Critical Sampling Rate Threshold While noise injection and electrode dropout represented controlled forms of signal degradation that could improve model generalization, our experiments with sampling rate reduction revealed a fundamental threshold below which EMG-based text decoding fails catastrophically. This dramatic performance cliff contrasts sharply with the more graceful degradation observed in electrode channel reduction experiments, where 12 electrodes per band still permitted functional decoding (52 CER). This finding suggests that temporal resolution is even more critical than spatial resolution for this task.

4.2 Optimal Signal Processing Parameters

Our experiments demonstrate that EMG data augmentation follows a clear non-linear response curve: too little augmentation risks overfitting, moderate augmentation provides optimal regularization, and excessive augmentation corrupts essential signal characteristics. The 0.5% noise level and 10% dropout probability represents a more appropriate balance for this particular EMG decoding task.

The moderate augmentation in table 1 likely provided improved noise robustness, enhanced fault tolerance by forcing redundant representations across multiple electrodes, and effective regularization. In contrast, table 2's performance degradation likely resulted from reduced signal-to-noise ratio, spatial information loss, and increased character confusion between similar muscle activation patterns.

4.3 Temporal Context Window Optimization

Our counter-intuitive finding regarding window size suggests that for sEMG-based keystroke prediction, smaller, more focused temporal windows may allow the model to better isolate and interpret the specific signal patterns associated with individual keystrokes, while larger windows might introduce confounding patterns or noise that complicate the learning process. These results highlight the critical importance of appropriate temporal context sizing in sequence modeling tasks involving physiological signals, where the relevant information may exist within more concentrated timeframes than initially assumed.

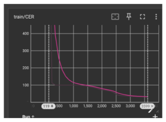
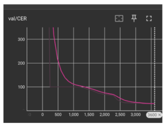
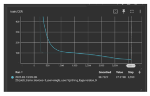
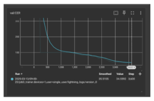
4.4 Architecture Selection for sEMG Data

While the TDS+Transformer approach offered theoretical advantages for modeling long-range dependencies, the TDS+GRU hybrid model proved more robust for preserving temporal dependencies within sequential data such as emg2qwerty. This suggests that for this specific task and dataset size, the sequential modeling capabilities of GRU networks are more suitable than the global attention mechanisms of transformers. It is possible that transformers would perform better with larger datasets that could fully leverage the advantages of self-attention mechanisms.

4.5 Limitations and Future Work

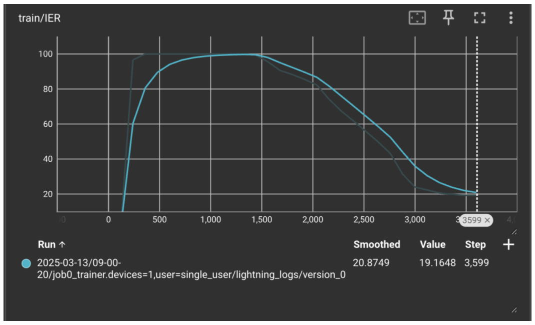
Due to time limitations, our experiments were mostly conducted independently (channel size, training data size, sampling rate, data preprocessing, and architectural changes) rather than iteratively finding the best configurations to maximize outperforming the baseline. As a result, we were not able to fully delve into understanding all cross-relationships and interactions.

Future work can include systematically exploring interactions between the variables we explored and training models on more than one subject to improve generalizability. There is great potential in the use of deep learning models for decoding sEMG signals, and further exploration and optimization would further improve accuracy and generalization.

	Training	Testing
Configuration 1		
Configuration 2		

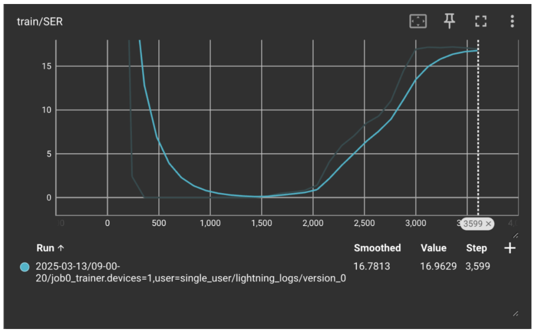
Configuration 1 & 2 Training + Validation CER Data

Figure 4: Visual Results for Table 2



Run ↑	Smoothed	Value	Step
2025-03-13/09-00-20/job0_trainer.devices=1,user=single_user/lightning_logs/version_0	20.8749	19.1648	3,599

Figure 5: train IER Pre-processing



Run ↑	Smoothed	Value	Step
2025-03-13/09-00-20/job0_trainer.devices=1,user=single_user/lightning_logs/version_0	16.7813	16.9629	3,599

Figure 6: train SER Pre-Processing

References

Roberto Merletti, Silvia Muceli, and Dario Farina. Surface electromyography: Barriers limiting widespread use of sEMG in clinical assessment and neurorehabilitation. *Frontiers in Neurology*, 12:738197, 2021. doi: 10.3389/fneur.2021.738197.

Garvit Vijayvargiya, Senthilkumar Mukherjee, et al. Deep learning methods for sEMG-based hand gesture classification: A comprehensive review. *Biomedical Signal Processing and Control*, 74: 103468, 2022. doi: 10.1016/j.bspc.2021.103468.