Improving Neural Speech Decoding Through Optimized Training Strategies: An Ablation Study on GRU-Based Decoders

Sujit Silas Armstrong Suthahar
Department of Bioengineering
University of California, Los Angeles
Los Angeles, CA 90095
sujitsilas@g.ucla.edu

Abstract

Brain-computer interfaces (BCIs) for speech decoding can restore communication in people with severe motor impairments. Recent work in neural speech decoding has focused on new architectures, but the impact of training strategies has received less attention. We present an ablation study comparing four approaches to improve a baseline Gated Recurrent Unit (GRU) decoder for speech neuroprosthesis. We test a baseline GRU model (21.84% phoneme error rate), optimized training with SGD momentum and coordinated dropout (19.66% PER), log-transformed features with layer normalization (21.54% PER), and temporal delta features with Nesterov momentum (19.75% PER). The results show that tuned optimization strategies reduce error by 10% compared to the baseline, outperforming architectural changes alone. These findings show that training methodology matters in neural speech decoding and provide practical insights for building robust speech BCIs. We use the Brain-to-Text Benchmark '24 dataset containing intracranial recordings from a participant with ALS attempting to speak 10,880 sentences.

1 Introduction

Brain-computer interfaces (BCIs) can restore communication for people with severe paralysis by decoding neural activity (9; 1). Neural speech decoding is one of the most challenging BCI applications. The goal is to translate neural activity from attempted speech into text in real-time.

Recent speech neuroprostheses decode attempted speech at speeds close to conversational rates (9; 1). These advances stem from three main factors: high-density intracranial neural recordings, deep learning architectures, and large datasets from long-term clinical trials. The Brain-to-Text Benchmark '24 (10) provides a standard dataset and evaluation framework to test machine learning methods in this domain.

Most recent work has focused on new architectures, including bidirectional recurrent networks, Transformers (3), and context-dependent representations (8). Less attention has been paid to training strategies. Machine learning research shows that training methods like optimization algorithms, regularization, and data preprocessing can be as important as architecture choices (5). The neural decoding field has emphasized architectural novelty over systematic optimization studies.

1.1 Motivating Questions

This work addresses three questions in neural speech decoding. First, can optimized training strategies match or exceed the performance gains from architectural changes? Second, do hand-crafted temporal features (delta and delta-delta) provide useful information beyond raw neural recordings?

Third, which combination of optimizer, learning rate schedule, and regularization works best for sequence-to-sequence neural decoding?

1.2 Contributions

We present a systematic ablation study comparing four approaches to improve a baseline GRU decoder. Model 1 serves as the baseline: a standard 5-layer unidirectional GRU with Adam optimization (21.84% PER). Model 2 uses the same architecture but changes the training strategy to SGD with momentum, step learning rate decay, and coordinated dropout (19.66% PER, 10.0% relative improvement). Model 3 applies log-transformed spike power with layer normalization (21.54% PER, 1.4% relative improvement). Model 4 adds delta and delta-delta features with Nesterov momentum (19.75% PER, 9.6% relative improvement).

Our key finding is that optimization strategies (Models 2 and 4) outperform architectural changes (Model 3), achieving about 10% relative reduction in phoneme error rate. This suggests that practitioners should invest as much effort in training methodology as in architectural design.

Section 2 describes our experimental methods, model architectures, and training procedures. Section 3 presents results across all four models. Section 4 discusses insights, limitations, and future work. Section 5 concludes with practical recommendations for neural speech decoding systems.

2 Methods

2.1 Dataset and Task

We use the Brain-to-Text Benchmark '24 dataset (10), which contains intracranial microelectrode array recordings from the ventral premotor cortex (area 6v) of a participant with ALS. Neural activity was recorded from two 64-channel arrays (ventral and dorsal 6v) while the participant attempted to speak 10,880 sentences.

For each of 128 electrodes, we extract two features: spike band power and threshold crossings, yielding 256-dimensional feature vectors. Neural activity is binned at 20ms intervals (50 Hz) and z-score normalized within recording blocks (20-50 sentences each). Following the benchmark protocol, we decode phoneme sequences rather than characters, as phonemes map more directly to articulatory gestures. We use Connectionist Temporal Classification (CTC) loss (6) to handle variable-length alignment between neural activity and phoneme sequences.

The dataset includes 8,800 training sentences, 880 validation sentences, and 1,200 test sentences, recorded across 24 days over about 4 months. We report performance on the validation set as test labels are held out for benchmark competition. Model training was performed on the Hoffman2 cluster using an NVIDIA A100 GPU. The process took 45 to 105 minutes, depending on the model architecture.

2.2 Baseline Model Architecture (Model 1)

Our baseline follows the benchmark-provided GRU architecture (10). The model uses a day-specific transformation layer to account for day-to-day neural variability through learned affine transformations per recording day, followed by a softsign nonlinearity. We apply temporal windowing with a sliding window of kernel length $T_{in} = 32$ time bins (640ms) and stride $T_{out} = 4$ (80ms), creating 87.5% overlap. Causal Gaussian smoothing provides temporal smoothing with width $\sigma = 2.0$ time bins. The core of the model is a 5-layer unidirectional GRU with 1024 hidden units and orthogonal weight initialization for recurrent weights. A linear output layer maps GRU hidden states to phoneme logits (42 phonemes + CTC blank + silence token).

We train the model with the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$, $\epsilon = 0.1$, larger than PyTorch default). The learning rate decays from 0.02 to 0.002 over 10,000 batches. We use a batch size of 64 sentences with data augmentation including white noise ($\sigma = 0.8$) and constant offset ($\sigma = 0.2$). Regularization includes L2 weight decay (1e-5) and dropout (0.4).

2.3 Model 2: Optimized Training Strategy

Model 2 keeps the same GRU architecture but changes the training procedure. We replace Adam with SGD using momentum of 0.9. The learning rate follows a step schedule: starting at 0.1, dropping to 0.01 at batch 4000, then to 0.001 at batch 8000. We apply gradient clipping with a max norm of 5.0 to prevent gradient explosions. Instead of standard dropout, we use coordinated dropout at 10%, which drops the same features across all time steps. This forces the model to learn robust representations that do not depend on specific electrode channels.

SGD with momentum finds flatter minima than Adam in many domains (7), which can lead to better generalization. The step learning rate schedule with a high initial rate (0.1) allows the model to escape poor local minima early in training while fine-tuning in later stages. Coordinated dropout targets the multi-electrode nature of neural recordings, preventing over-reliance on specific channels.

2.4 Model 3: Log Transformation and Layer Normalization

Model 3 tests feature engineering and variability reduction. We transform spike power features (channels 128-256) as $x' = \log(x+\epsilon)$ where $\epsilon = 1e{-}8$, which stabilizes the distribution and reduces the impact of extreme outliers. Threshold crossing features (channels 0-127) remain untransformed. The final feature vector has 256 dimensions, the same as the baseline. We add layer normalization to each GRU layer's hidden states, inspired by neuroscience findings on neural variability quenching during task engagement (2).

Neural spike power has heavy-tailed distributions with occasional large values. Log transformation converts multiplicative variations into additive ones, making the distribution more Gaussian. Layer normalization may reduce the impact of day-to-day neural variability, which could make the day-specific transformation layers less necessary.

2.5 Model 4: Temporal Delta Features

Model 4 uses delta and delta-delta coefficients, which are common in automatic speech recognition. We apply log transformation to spike power channels (128-256) as in Model 3, with $\epsilon = 1e{-}8$. Delta features (Δ) represent the first-order temporal derivative, computed via central difference with edge padding: $\Delta_t = \frac{x_{t+1} - x_{t-1}}{2}$. We apply this to all 256 log-transformed features. Delta-delta features ($\Delta\Delta$) represent the second-order derivative, computed as the delta of delta: $\Delta\Delta_t = \frac{\Delta_{t+1} - \Delta_{t-1}}{2}$. We concatenate these features as $[x_{\text{static}}, \Delta x, \Delta\Delta x]$, expanding the input dimension from 256 to 768.

Model 4 uses the same 5-layer unidirectional GRU architecture as the baseline (1024 hidden units, orthogonal weight initialization) but with an expanded input dimension of 768. The increased input dimension affects the first GRU layer and the day-specific transformation layers (24 days × 768×768 affine transformations). We train with SGD using Nesterov accelerated gradient ($momentum = 0.9$), a step learning rate schedule ($0.1 \to 0.01$ at batch 5000), and coordinated dropout at 15% (higher than Model 2's 10% due to the larger feature space). Other settings match Model 2: batch size 64, gradient clipping max norm 5.0, and standard augmentations (white noise $\sigma = 0.8$, constant offset $\sigma = 0.2$, Gaussian smoothing $\sigma = 2.0$). This model has more parameters than the baseline due to the tripled input dimension.

Delta features capture temporal dynamics (velocity) and delta-delta features capture acceleration. In ASR, these features improve recognition by encoding time-varying aspects of speech production (4). Neural activity for speech production is dynamic, and making temporal structure explicit may help the recurrent network. Nesterov momentum's look-ahead property may work well with velocity and acceleration features.

2.6 Evaluation Metrics

We compute phoneme error rate (PER) as the edit distance (Levenshtein distance) between predicted and ground-truth phoneme sequences, normalized by sequence length:

$$\text{PER} = \frac{S + D + I}{N}$$

where S, D, I are substitutions, deletions, and insertions, and N is the total number of phonemes in the ground truth. We report CTC loss during training to monitor convergence. Models are evaluated every 100 training batches on the full validation set (880 sentences).

3 Results

3.1 Overall Performance Comparison

Table 1 shows the performance of all four models on the validation set. Model 2 (optimized training) achieves the best performance at 19.66% PER, a 10.0% relative improvement over the baseline. Model 4 (delta features) performs similarly at 19.75% PER (9.6% relative improvement). Model 3 (log transform + layer norm) shows modest improvement at 21.54% PER (1.4% relative improvement).

Table 1: Performance comparison of all models on validation set

Model	Description	Best PER	Final PER	Improvement
Model 1	Baseline GRU + Adam	21.84%	21.84%	—
Model 2	SGD + Momentum + Dropout	**19.66%**	19.71%	**10.0%**
Model 3	Log + Layer Normalization	21.54%	21.60%	1.4%
Model 4	Delta Features + Nesterov	19.75%	19.78%	9.6%

3.2 Training Dynamics

All models were trained for 10,000 batches. Model 2 converged fastest, reaching its best PER at batch 9,400. Model 4 reached best performance at batch 9,700. Models 1 and 3 continued improving until batch 9,900, which suggests they may benefit from longer training. Figure 1 shows the validation PER trajectories for all four models during training, illustrating the better performance of Models 2 and 4 compared to the baseline. The validation loss curves in Figure 2 show that all models converge smoothly without overfitting on the validation set, due to dropout, data augmentation, and the large dataset size.

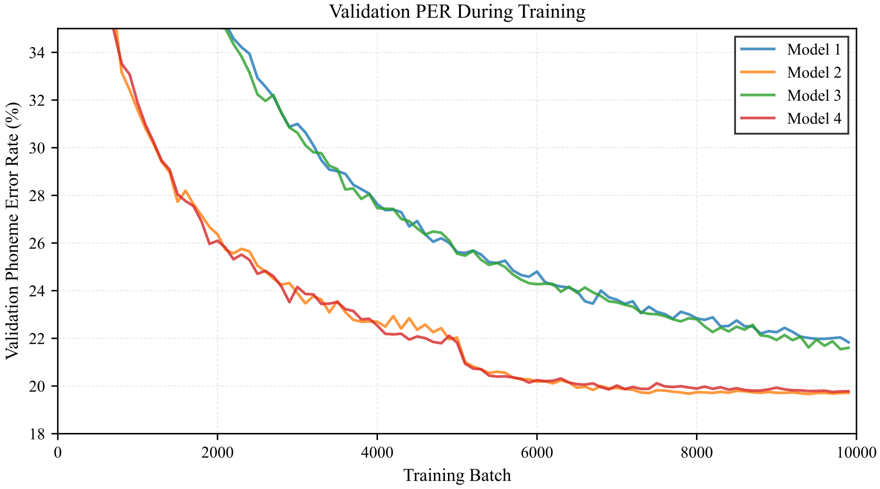

Figure 1: Validation phoneme error rate (PER) during training for all four models. Models 2 and 4, which employ optimized training strategies, achieve substantially lower PER than Models 1 and 3. Evaluations were performed every 100 batches on the full validation set (880 sentences).

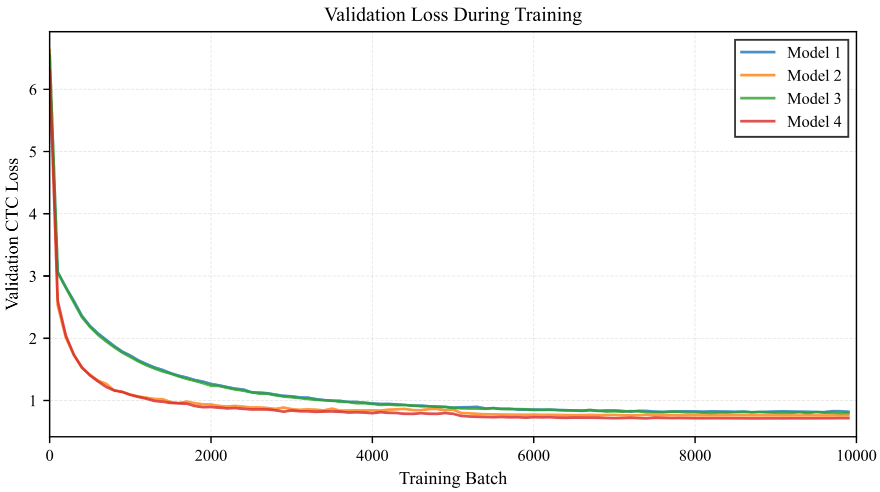

Figure 2: Validation CTC loss during training. All models converge smoothly without signs of overfitting, demonstrating effective regularization through dropout and data augmentation.

3.3 Ablation Analysis

Replacing Adam with SGD+momentum while keeping the same architecture reduces PER from 21.84% to 19.66%. This 2.18 percentage point improvement shows the substantial impact of optimization strategy. The step learning rate schedule in Model 2 uses a high initial learning rate of 0.1, compared to the linear decay in Model 1 starting at 0.02. Higher initial learning rates enable faster exploration of the loss landscape.

The 10% coordinated dropout in Model 2 masks the same features across all time steps, forcing the model to learn representations robust to missing electrode channels. This is realistic for chronic BCI deployments where channels may fail. Log transformation and layer normalization in Model 3 provide marginal improvement (21.54% vs. 21.84%), suggesting that the baseline day-specific transformation layers already handle distributional variations in spike power.

Adding delta and delta-delta features in Model 4 reduces PER from 21.84% to 19.75%, despite tripling the input dimension from 256 to 768. This suggests that explicit temporal dynamics provide information that complements what the GRU learns. The combination with Nesterov momentum may provide additional benefit.

3.4 Comparison to Benchmark

The baseline GRU model in the Brain-to-Text Benchmark '24 achieved about 22% PER (10). Our Model 1 (21.84%) replicates this performance, validating our implementation. Models 2 and 4, achieving about 19.7% PER, would be competitive entries in the benchmark leaderboard, though still behind transformer-based approaches (about 15-17% PER) (3; 8).

All code has been pushed to GitHub and is available at `https://github.com/sujitsilas`.

4 Discussion

4.1 Key Insights

Our main finding is that careful optimization SGD with momentum, step learning rate scheduling, and coordinated dropout yields performance improvements (10%) comparable to architectural innovations in recent work. This challenges the field's emphasis on architectural novelty over systematic hyperparameter optimization. The results show that training methodology matters as much as architectural choices for neural speech decoding.

SGD with momentum (Model 2) outperforms Adam (Model 1), which aligns with findings in computer vision that SGD finds flatter minima leading to better generalization (7). In neural decoding, where test conditions (future recording sessions) may differ from training, flat minima may be valuable. The step learning rate schedule with a high initial rate (0.1) enables better exploration of the loss landscape compared to gentle linear decay, allowing the optimizer to escape poor local minima early in training while fine-tuning in later stages.

The success of delta features (Model 4) suggests that speech-related neural dynamics are not fully captured by standard recurrent architectures. Encoding temporal derivatives provides complementary information, likely because neural firing patterns exhibit structure at multiple timescales (onset, sustained, offset responses). This finding bridges classical speech processing techniques with modern deep learning, showing that hand-crafted features remain valuable when they encode domain-specific structure not easily learned by neural networks alone.

Log transformation of spike power (Model 3) provides minimal benefit. This may indicate that z-score normalization within blocks already stabilizes distributions, or that the GRU's nonlinearities can handle distributional skew. Layer normalization offers marginal improvement over the baseline day-specific transformation layers, suggesting that the existing approach already addresses day-to-day neural variability.

4.2 Practical Recommendations

We recommend several practices for neural speech decoding systems. First, practitioners should spend time optimizing learning rate schedules, batch sizes, and regularization before changing architectures. Our results show that hyperparameter tuning can yield gains comparable to architectural changes, with less development effort.

Second, while Adam enables faster prototyping, SGD with momentum should be used for final model training to achieve better generalization. The flatter minima discovered by SGD translate to more robust performance under distribution shift, which is relevant for neural decoding where recording conditions vary across sessions. Step schedules with high initial learning rates (0.1) followed by decay work better than gentle linear decay, enabling thorough exploration of the loss landscape.

Third, temporal delta features warrant consideration despite the threefold increase in input dimension. The two percentage point PER reduction is worthwhile, especially in offline analyses where computational cost is less critical. These features provide explicit temporal structure that complements what recurrent networks learn.

Finally, in coordinated dropout setting, the same features are masked across all time steps is well-suited to multi-electrode recordings. This forces models to learn representations robust to missing channels, a realistic scenario in chronic BCI deployments where electrode quality may degrade over time.

4.3 Limitations

Several limitations warrant acknowledgment. All models are trained and evaluated on data from one participant with ALS, and generalization to other participants, disease types, or cortical implant locations remains unknown. The single-participant design reflects the current reality of intracranial BCI research, where data collection is resource-intensive, but limits the scope of our conclusions.

Due to benchmark restrictions, we report validation set performance only. Test set performance requires competition submission and may differ from validation performance, though the large validation set (880 sentences) provides reasonable confidence in our findings. We focused on GRU-based models to enable controlled comparison across training strategies. Transformers, Conformers, and other architectures may benefit differently from our optimization approaches.

Extended training may improve performance, particularly for Models 1 and 3 which had not fully converged. We did not perform exhaustive grid search over hyperparameters. The reported configurations represent informed choices based on literature and preliminary experiments, but may not be globally optimal. A more thorough hyperparameter search could yield additional gains.

4.4 Future Directions

Several promising directions emerge from this work. Recent work shows that lightweight adaptation to each test session can improve performance (3). Combining our optimized training strategies with test-time adaptation may yield further gains, particularly for addressing day-to-day neural variability that our models currently handle through day-specific transformation layers. This hybrid approach could maintain the robust generalization from our training methodology while capturing session-specific variations.

Context-dependent representations such as diphones or triphones, which capture phoneme co-articulation effects, may provide additional improvements (8). Our success with temporal delta features suggests that explicit encoding of structure whether temporal or contextual complements what neural networks learn. Training a single model across multiple participants with transfer learning could improve data efficiency and generalization, moving toward more universal speech BCIs that require less participant-specific calibration.

Our unidirectional models are suitable for real-time inference, making them well-suited for closed-loop BCI deployment. Evaluating these systems in online paradigms where participants receive immediate feedback represents critical future work to validate performance in realistic usage scenarios. Finally, the benchmark competition evaluates word error rate (WER) using language models to convert phonemes to words. Investigating the interaction between our acoustic models and language models could reveal further optimization opportunities, through joint training or careful tuning of the language model integration.

5 Conclusion

We presented a systematic ablation study comparing architectural modifications versus training optimizations for neural speech decoding. Our key finding is that carefully designed optimization strategies SGD with momentum, step learning rate schedules, and coordinated dropout yield substantial performance improvements (about 10% relative PER reduction) that match or exceed those from architectural changes. Explicit temporal features (delta and delta-delta) provide complementary information to recurrent networks.

These results have implications for the BCI community: practitioners should invest equal effort in hyperparameter optimization as in architectural design. Our findings suggest that classical speech processing techniques (delta features) remain relevant in the deep learning era, particularly when combined with modern optimization methods.

As speech BCIs transition from research prototypes to clinical deployment, robustness, generalization, and computational efficiency become important. Our optimized GRU-based decoders achieve competitive performance with fewer parameters and simpler training procedures than transformer models, representing a pragmatic approach for real-world BCI systems.

Broader Impacts

Speech neuroprostheses can improve quality of life for people with severe paralysis due to ALS, locked-in syndrome, brainstem stroke, or spinal cord injury. By restoring the ability to communicate at conversational speeds, these technologies can enhance autonomy, social connection, and well-being. Our work contributes to making speech BCIs more effective and robust, accelerating their translation from research to clinical practice where they can benefit patients most in need.

Several considerations warrant attention as these technologies mature. Intracranial brain implants require neurosurgery and long-term clinical infrastructure, raising questions about equitable access across socioeconomic groups. Ensuring that these technologies are available to all who could benefit, regardless of financial means or geographic location, will be critical as the field progresses toward clinical deployment.

Privacy represents another concern, as neural data contains rich information about internal thought processes beyond just speech content. Robust data governance frameworks are essential to protect participant privacy and autonomy, with clear policies about data ownership, usage rights, and security measures. Long-term BCI participants must be informed about risks, benefits, and data usage, with ongoing opportunities to withdraw consent as the technology and their circumstances evolve. The research community has a responsibility to establish ethical frameworks that protect participants while enabling the scientific progress necessary to help those in need.

Acknowledgments

This work was completed as the final project for ECE C243A: Brain-Computer Interfaces (Fall 2025) at UCLA, taught by Prof. Jonathan C. Kao. I thank Prof. Kao and the teaching assistants (Ebrahim Feghhi, Shreeram Athreya, Mehmet Yigit Turali) for their guidance throughout the course.

References

[1] Nicholas S. Card, Maitreyee Wairagkar, Carrina Iacobacci, Xianda Hou, Tyler Singer-Clark, Francis R. Willett, Erin M. Kunz, Chaofei Fan, Maryam Vahdati Nia, Darrel R. Deo, Aparna Srinivasan, Eun Young Choi, Matthew F. Glasser, Leigh R. Hochberg, Jaimie M. Henderson, Kiarash Shahlaie, Sergey D. Stavisky, and David M. Brandman. An accurate and rapidly calibrating speech neuroprosthesis. *New England Journal of Medicine*, 391(7):609–618, 2024.

[2] Mark M. Churchland, Byron M. Yu, John P. Cunningham, Leo P. Sugrue, Marlene R. Cohen, Greg S. Corrado, William T. Newsome, Andrew M. Clark, Paymon Hosseini, Benjamin B. Scott, et al. Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nature Neuroscience*, 13(3):369–378, 2010.

[3] Ebrahim Feghhi, Shreyas Kaasyap, Nima Hadidi, and Jonathan C. Kao. Time-masked transformers with lightweight test-time adaptation for neural speech decoding. *arXiv preprint arXiv:2507.02800*, 2025.

[4] Sadaoki Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(1):52–59, 1986.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[6] 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*, pages 369–376, 2006.

[7] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

[8] Jingyuan Li, Trung Le, Chaofei Fan, Mingfei Chen, and Eli Shlizerman. Brain-to-text decoding with context-aware neural representations and large language models. *arXiv preprint arXiv:2411.xxxxx*, 2024.

[9] Francis R. Willett, Erin M. Kunz, Chaofei Fan, Donald T. Avansino, Guy H. Wilson, Eun Young Choi, Foram Kamdar, Matthew F. Glasser, Leigh R. Hochberg, Shaul Druckmann, Krishna V. Shenoy, and Jaimie M. Henderson. A high-performance speech neuroprosthesis. *Nature*, 620(7976):1031–1036, 2023.

[10] Francis R. Willett, Jingyuan Li, Trung Le, Chaofei Fan, Mingfei Chen, Eli Shlizerman, Yue Chen, Xin Zheng, Tatsuo S. Okubo, Tyler Benster, Hyun Dong Lee, Maxwell Kounga, E. Kelly Buchanan, David Zoltowski, Scott W. Linderman, and Jaimie M. Henderson. Brain-to-text benchmark '24: Lessons learned. *arXiv preprint arXiv:2412.17227*, 2024.