Deep learning (INFOMDLR) - Assignment 2

Mahshid Jafar Tajrishi
Graduate School of Natural Sciences
Utrecht University
Netherlands
m.jafartajrishi@students.uu.nl

Bar Melinarskiy
Graduate School of Natural Sciences
Utrecht University
Netherlands
b.melinarskiy@students.uu.nl

Cis van Aken
Graduate School of Natural Sciences
Utrecht University
Netherlands
c.j.f.vanaken@students.uu.nl

Simon van Klompenburg
Graduate School of Natural Sciences
Utrecht University
Netherlands
a.s.vanklompenburg@students.uu.nl

***Abstract*—Magnetoencephalography (MEG) provides high-resolution neuroimaging data that is crucial for understanding cognitive processes and diagnosing neurological conditions. However, the complexity and noise inherent in MEG data pose significant challenges for reliable classification. In this study, we design and implement a deep learning model to classify four cognitive tasks (resting, math, memory, and motor) from MEG recordings. Inspired by Augmented Attention-MultiviewNet, our architecture incorporates parallel spatial and temporal processing branches, self-attention mechanisms, and extensive data augmentation strategies. We evaluate performance under intra-subject and cross-subject settings. The intra-subject model achieved perfect classification accuracy (100%) across all tasks, demonstrating the capability of the architecture when training and testing on the same participant. However, cross-subject generalization proved challenging, with overall test accuracy reaching only 70.83%, improving modestly to 75.00% with self-attention mechanisms. Principal Component Analysis revealed that test samples formed separate clusters from training samples in memory and motor tasks, suggesting insufficient training data diversity as the primary limitation rather than architectural constraints. Our findings highlight the fundamental challenges of generalizing across individuals in neuroimaging and emphasize the critical importance of dataset representativeness over model complexity. These insights contribute to the development of more robust and generalizable MEG classification models suitable for clinical and cognitive neuroscience applications.**

***Index Terms*—MEG, Brain Signal Analysis, Deep Learning, Classification**

1. Introduction

Magnetoencephalography (MEG) is one of the most advanced neuroimaging techniques available for mapping both temporal and spatial brain activity, making it invaluable in cognitive neuroscience and clinical diagnosis. However, the high dimensionality and inherent noise in MEG signals present significant challenges for data analysis and interpretation.

In this paper, we present the design and implementation of a deep learning model for MEG-based task classification. Our model analyses MEG recordings to predict which cognitive task a subject was performing during data acquisition, including resting state, mathematical and language processing tasks, working memory tasks, and motor tasks. This classification approach aims to improve signal decoding and pattern recognition capabilities, thereby enhancing non-invasive brain monitoring techniques.

Accurate MEG task classification has important real-world applications in medical settings, including earlier detection of neurological diseases such as epilepsy and Alzheimer's, and monitoring treatment effectiveness [1, 2]. By leveraging deep learning's capacity to identify complex patterns in high-dimensional data, beyond what is achievable through traditional statistical methods, this study contributes toward more precise and scalable approaches for analyzing brain signal data.

2. Related Work

As mentioned before, the analysis and classification of MEG data is not a new task. Plenty of papers on the topic have been written and plenty of models have been created, as it potentially provides a lot of medical utility. For instance, Zheng et al developed EMS-net (epileptic MEG-spikes network) to detect epileptic spikes. EMS-net is a multiview deep learning network, that manages to achieve high accuracy in epileptic spike detection. While the exact architecture is not open source, it is a multiview model, suggesting that it feeds MEG data through different channels before combining the outputs for final classification[3].

Another paper that served an inspiration for this research, was written by abdellaoui et al. In this paper, they propose and rigorously test three attention-augmented deep learning architectures designed specifically for classifying MEG signals[4].

- **AA-EEGNet** A convolutional model derivative of EEG-Net, adapted for MEG data and enhanced with both self-attention and global attention mechanisms.
- **AA-CascadeNet** A hybrid model combining convolutional layers with LSTM modules in a cascade configuration, also enhanced with the dual attention schemes.
- **AA-MultiviewNet** A dual-stream architecture that fuses spatial (convolutional) and temporal (LSTM) features before applying attention to the combined representation.

Each model incorporates self-attention to distinguish intra-sequence interactions and global attention, to detect the most informative features. This paper shows that the correct and targeted implementation of both self and global attention can be beneficial for the model accuracy.

From these papers, we learnt that a lightweight convolutional backbone is still a very useful baseline for MEG decoding; however, for higher accuracies, more powerful supplementary architectures are needed. In addition, the processing and extraction of spatial and temporal features sequentially, before integrating and classifying, boosts performance and efficiency. Finally, we may look to implement attention mechanisms into our own model to boost performance and increase model robustness.

3. Data

The dataset used in this study consists of MEG recordings from seven participants across four distinct experimental conditions. Model performance was assessed through two classification approaches: intra-subject classification, where the model was trained and tested on data from the same participant, and cross-subject classification, where the model was trained on data from two participants and tested on data from the remaining four participants.

The four experimental conditions were as follows:

- **Resting Task:** Participants maintained a relaxed resting state with minimal cognitive load
- **Math & Story Task:** Participants performed mental calculations and language processing tasks
- **Working Memory Task:** Participants completed a memorization task.
- **Motor Task:** Participants executed a motor task, moving fingers or feet.

Each participant-task combination produced 8 MEG data matrices. The spatial dimension consisted of 248 magnetometer sensors, positioned on the human scalp. The recordings were made at a sample rate of 2034 Hz and lasted approximately 17.5 seconds, resulting in a total of 35,624 time points per recording. The data matrices thus had dimensions: 248 x 35,624 (sensors x time points).

3.1. Normalization

The order of magnitude of this data is 10e-15 Tesla, which is not suitable for the training of neural networks. Furthermore, inspection revealed consistently anomalous values in sensor channel 236 across all recordings. Initial application of min-max normalization (scaling all data to [0,1] range) solved the first issue, but the second persisted. Standard z-score normalization also proved insufficient as the outlier channel inflated variance estimates, compromising normalization effectiveness. To address this, we instead implemented temporal Z-score normalization. That is, for every value $x_{i,j}$ across all matrices, values were normalized as: $z_{i,j} = \frac{x_{i,j} - \mu_{i,j}}{\sigma_{i,j}}$ where $\mu_{i,j}$ and $\sigma_{i,j}$ represent the mean and standard deviation of sensor i at time point j across all matrices. This approach preserved relative activation patterns, while also standardizing amplitude scales across sensors and time points.

3.2. Downsampling

Following established neuroscience practice, temporal downsampling was applied to reduce computational complexity [5]. Specifically, average pooling over consecutive time points was implemented using a window size of 7, meaning that every 7 consecutive time points were averaged into a single value. This window size was chosen as a balance between reducing training times while maintaining high accuracy. Alternative window sizes in the range of 5-15 were tested and achieved similar results, confirming that a moderate downsampling rate provides optimal performance. In contrast, a larger window size of 25 resulted in lower performance, likely due to excessive information loss.

4. Model

4.1. Model Overview

Our model architecture builds upon the AA-MultiviewNet framework proposed by Alaoui Abdellaoui et al. for MEG-based brain state classification, incorporating multiview attention mechanisms for enhanced task classification [4]. Both models employ a dual-stream design in which MEG signals are processed in parallel through a spatial convolutional branch and a temporal LSTM-based branch. Each stream independently extracts features from multiple time windows, and their representations are fused to perform multi-class classification.

As illustrated in Fig. 1, our implementation retains the core architectural design of the original model but introduces several important modifications. The original AA-MultiviewNet, implemented in Keras, applies self-attention only in the first convolutional layer, uses fixed (7×7) kernel sizes, omits padding to preserve spatial boundaries, and doubles the number of filters at each convolutional stage. In contrast, our PyTorch implementation supports configurable kernel sizes, paddings, and activation functions, and modularizes the self-attention mechanism to enable greater flexibility and extensibility.

A further distinction lies in the fusion strategy. In both models, CNN outputs from each window are independently processed and summed across time. However, in our version, per-window dense layers are applied in both the CNN and LSTM streams prior to fusion, offering finer control over representation shaping. The final spatial and temporal embeddings are then concatenated and passed to a classifier. By comparison, the original model concatenates the LSTM output with the summed CNN output directly after the final LSTM layer, without further stream-specific transformation.

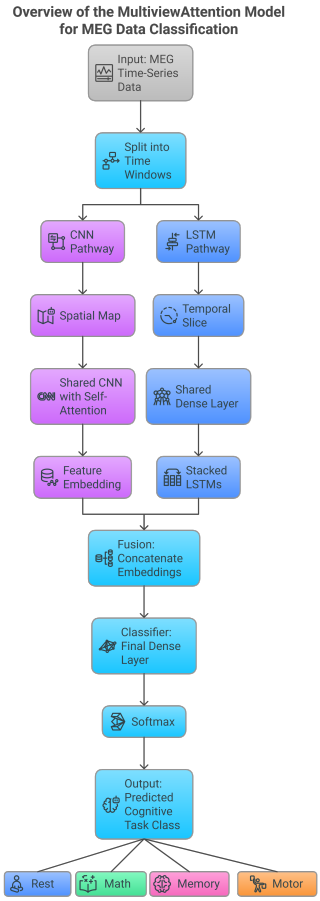

Fig. 1: Overview of the Attention Augmented Multiview Model for MEG Data Classification. The model processes MEG time-series data by first segmenting it into multiple time windows. Each window is passed through two parallel pathways: a spatial CNN branch with integrated self-attention, and a temporal LSTM branch. The CNN pathway extracts spatial embeddings from each window's mesh-structured representation, while the LSTM pathway captures temporal dependencies across time. The resulting embeddings from both branches are fused and passed through a final classifier to predict the cognitive task category (resting, math, memory, or motor).

Building upon the original self-attention framework, we expanded the attention mechanism implementation to explore potential performance improvements, as will be discussed in Section 5.2. Our *AugmentedConv2D* module processes each spatial input through both a standard convolutional path and a self-attention branch, concatenating their outputs to capture both local spatial patterns and long-range spatial dependencies. The self-attention component employs multi-head attention with relative positional embeddings, allowing the model to attend to spatially distant MEG channels while maintaining awareness of their topological relationships on the sensor array.

Additionally, we extended the original implementation by incorporating commonly used data augmentation techniques to improve the model's generalization and robustness when working with MEG signals [6]. Specifically, we apply Gaussian noise injection, random temporal shifts, amplitude scaling, and channel dropout during training to simulate variability and reduce overfitting. These augmentations are implemented in a custom PyTorch Dataset class that also handles input windowing and reshaping, resulting in a more flexible and resilient training pipeline.

Together, these modifications enhance flexibility, modularity, and adaptability to different experimental configurations while maintaining the conceptual foundation of the original AA-MultiviewNet architecture.

4.2. Hyperparameter Optimization

To optimize model performance, we tuned hyperparameters using four-fold stratified cross-validation on the training set, ensuring balanced task type distribution across folds [7]. For this purpose, Optuna [8] was used to automate hyperparameter optimization. Table I summarizes the hyperparameters that were tuned for each model, together with the possible values that were tested:

TABLE I: Chosen Hyperparameters for All Three Models with Possible Values

Parameter	Intra-Subject	Cross-Subject	Cross-Subject w/ Self Attention	Possible Values Tested
Window Size	8	8	4	{4, 8, 16, 24, 32, 48, 50}
Mesh (Rows×Cols)	8×31	8×31	43×23	Derived from valid combinations of depth, rows, and columns
Depth	1	1	248	{1, 2, 4, 8, 10, 248}
Conv1 (Filters, Kernel, Pad)	128, (5,5), 2	128, (5,5), 2	16, (5,5), 1	Filters: {128}, Kernel: {(3,3), (5,5)}, Pad: {0, 1, 2}
Conv2 (Filters, Kernel, Pad)	256, (3,3), 1	256, (3,3), 1	128, (7,7), 1	Filters: {128, 256}, Kernel: {(3,3), (5,5), (7,7)}, Pad: {0, 1, 2}
Conv3 (Filters, Kernel, Pad)	64, (5,5), 2	256, (3,3), 1	256, (3,3), 2	Filters: {64, 128, 256}, Kernel: {(3,3), (5,5), (7,7)}, Pad: {0, 1, 2}
Conv1 Activation	GELU	GELU	ELU	{ELU, ReLU, GELU, Leaky ReLU}
Conv2 Activation	Leaky ReLU	Leaky ReLU	Leaky ReLU	{ELU, ReLU, GELU, Leaky ReLU}
Conv3 Activation	ELU	ELU	ReLU	{ELU, ReLU, GELU, Leaky ReLU}
Dense (Nodes, Activation)	128, ReLU	128, GELU	256, Leaky ReLU	Nodes: {256}, Activation: {ReLU, GELU, Leaky ReLU}
LSTM1 Cells	32	64	32	{32, 64, 128}
LSTM2 Cells	64	64	256	{32, 64, 128, 256}
Dense3 (Nodes, Activation)	64, ReLU	32, GELU	32, GELU	Nodes: {32, 64, 128}, Activation: {ReLU, GELU, Leaky ReLU}
Learning Rate	0.001	0.0011	0.0042	[4e-4, 2e-3] (log scale)
Batch Size	8	4	8	{2, 8, 16, 32}
Dropout	0.10	0.05	0.15	[0.1, 0.25] (step=0.05)
Optimizer	Adam	Adam	RMSProp	{Adam, AdamW, RMSProp}
Weight Decay	8.69e-7	3.04e-5	2.04e-4	[1e-7, 1e-4] (log scale)
Attention (Heads, Layers)	–	–	2, 1	Heads: {2, 3, 4, 6, 8}, Layers: {1, 2}
Channels/Classes	248/4	248/4	248/4	Fixed: 248/4

Computational constraints limited the scope of our hyperparameter search, preventing extensive exploration of highly complex parameter combinations. However, our optimization trials revealed distinct patterns across model variants. The intra-subject model consistently achieved near-perfect performance across most parameter configurations, suggesting that this task is relatively straightforward given that training and testing occur on the same individual. In contrast, the cross-subject model exhibited substantial performance variation across different hyperparameter settings, indicating that effective generalization across subjects requires careful parameter tuning.

The heatmap, shown in Fig. 2, summarizes the validation accuracy results from our initial hyperparameter tuning trials for the cross-subject model. Each subplot highlights how a specific hyperparameter influenced performance across different cognitive tasks and overall accuracy.

A few patterns emerge across the heatmaps. The learning rate had a strong and consistent effect, with lower values leading to higher accuracy on most tasks. The dropout rate, on the other hand, showed a more nuanced influence: while low and intermediate rates offered sufficient overall performance, the resting task benefited most from a higher rate of 0.15. Batch size also played a role, with smaller values (4 and 8)

generally outperforming the larger size of 16, particularly in the resting and overall scores, which makes sense given our relatively small dataset.

Weight decay exhibits a pronounced peak at 0.0005, consistently delivering the highest accuracy across all tasks and the overall average, suggesting that a moderate degree of regularization is optimal in this setting. Finally, we also tuned parameters regarding the model's complexity - increasing the number of LSTM cells tends to improve accuracy. This effect is especially strong for the first LSTM layer, where the largest tested size (128) produces top results for the resting task. The second layer (LSTM2) shows a similar but slightly more subdued trend, with 64 and 128 cells both performing well across tasks.

These insights guided our choice of hyperparameter ranges for the subsequent Optuna-based optimization, allowing us to focus the search on the most promising configurations.

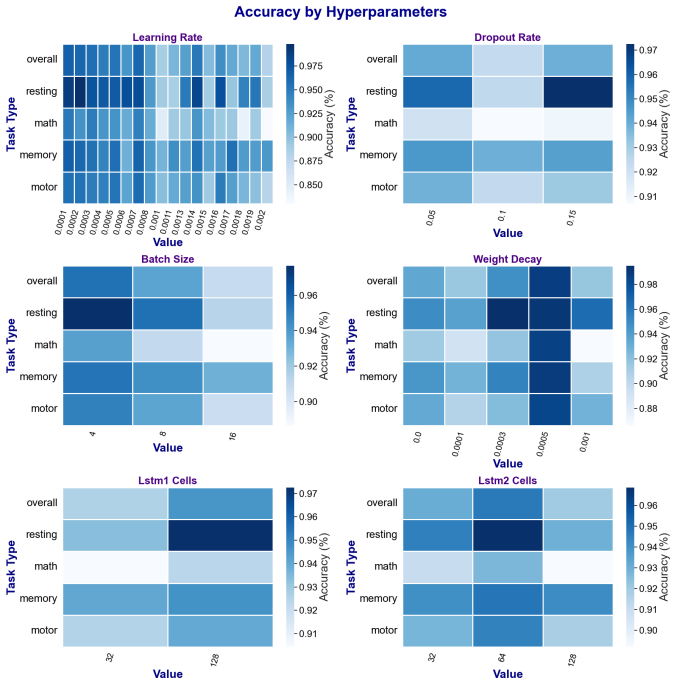

Fig. 2: Performance Across Hyperparameter Settings. Heatmaps showing model performance across different hyperparameter configurations for the cross-subject multiview model without self-attention. Results are based on 50 trials runs with varying hyperparameter combinations sampled from the parameter space defined in Table I. Note that hyperparameter exploration was conducted using Optuna's Tree-structured Parzen Estimator, which sequentially samples parameter combinations by balancing exploration of unexplored regions with exploitation of high-performing parameter neighborhoods [8].

5. Experimental Results

Following hyperparameter optimization, we trained each model on the complete training dataset corresponding to its evaluation scenario (intra or cross). Figure 3 presents a comparative analysis of test set performance between the intra-subject and cross-subject approaches across the four cognitive tasks, as well as overall classification accuracy.

5.1. Intra-Subject vs. Cross-Subject Classification Performance

In the intra-subject scenario, the model achieved perfect performance across all task types, reaching 100% accuracy for all of them both in the train and test sets. This high performance demonstrates that the model architecture possesses sufficient complexity to distinguish between the four cognitive states when trained and tested on the same participant. However, the cross-subject evaluation revealed a substantial performance gap, with overall test accuracy dropping to 70.83% compared to 98.44% training accuracy. This high performance on train combined with a considerable drop on test indicates significant overfitting, where the model learns participant-specific patterns rather than generalizable cognitive signatures. The task-specific performance degradation followed a clear pattern: while the *resting* state maintained perfect classification, active cognitive tasks showed progressively worse generalization. The *math* task achieved 83.33% accuracy, the *motor* task dropped to 58.33%, and the *memory* task performed poorest at 41.67% accuracy.

Notably, during hyperparameter optimization for the cross-subject model, we observed that window size significantly influenced task-specific performance. The *memory* task consistently achieved higher accuracy with larger window sizes (30+ time steps), potentially indicating that this cognitive state may require longer temporal context for accurate classification. In contrast, the *math* and *motor* tasks showed more stable performance across participants with moderate window sizes, suggesting that these tasks may be characterized by more temporally localized neural patterns.

These results show that the model is complex enough to classify all four tasks, given that it is trained on the same participant as it will be tested on (Intra-Subject classification). For Cross-Subject classification, the model performed worse, since it needs to learn to generalize to four participants, while only training on data from two different participants.

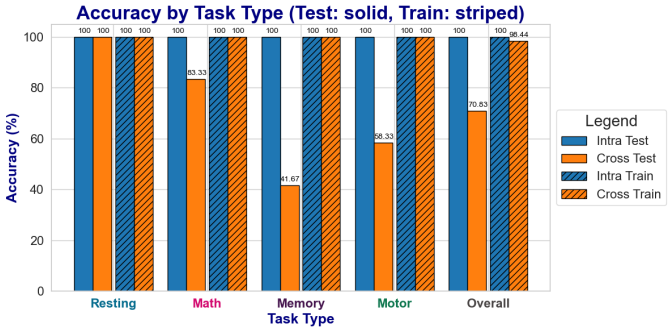

Fig. 3: Intra-Subject vs. Cross-Subject Classification Performance Across Task Type. In each subplot, the solid-colored bar represents test accuracies, while the striped bars correspond to training accuracies.

5.2. *Cross Model Improvements*

The substantial train-test performance gap observed in cross-subject classification (98.44% train vs. 70.83% test accuracy) indicated severe overfitting, suggesting that architectural modifications were needed to improve generalization. To address this limitation, we incorporated a self-attention mechanism into the AA-MultiviewNet architecture (as discussed in Section 4.1).

Fig. 4 illustrates the per-patient and overall accuracy for the four cognitive tasks in both test and training sets, using two model variants: AA-MultiviewNet without and with Self Attention. A significant discrepancy is observed between the training and test performance of the original model, particularly in the *math* and *memory* tasks. For example, in the baseline AA-MultiviewNet, the *memory* task reaches 100% accuracy on the training set but only 41.7% on the test set, highlighting severe overfitting. The *math* task also exhibits unstable test performance, with some patients scoring as low as 50%. Moreover, in both the *memory* and *motor* tasks, there are individual patients in the test set for whom the model achieved 0% accuracy, demonstrating a complete failure to generalize for those cases.

As shown in the bottom row of Fig. 4, the self-attention mechanism yielded mixed results. Overall test accuracy improved from 70.83% to 75.00%, with the *memory* task showing modest gains from 41.7% to 50%. The *math* task demonstrated more consistent performance across patients, reducing the extreme variability observed in the baseline model. However, these improvements came at the cost of increased model complexity, as evidenced by perfect training accuracy (100%) across all tasks, suggesting the model was fitting the limited training data even more precisely.

The results across different test sets in Table II reveal important insights about generalization patterns. Test Set 1 (patient 162935) showed dramatic improvement with self-attention (87.5% to 100%), while Test Set 2 (patient 707749) actually performed worse (43.75% to 37.5%). Test Set 3, containing two patients (735148 and 725751), showed modest gains (81.25% to 87.5%). This inconsistent pattern indicates that the self-attention mechanism's effectiveness varies significantly across individual patients, with some benefiting substantially while others experience reduced performance.

TABLE II: Cross-Subject Model Accuracy Results Across Test Sets. The test sets included the following patients: Test1 - 162935, Test2 - 707749, and Test3 - 735148, 725751.

Model	Accuracy Test1	Accuracy Test2	Accuracy Test3
AA-MultiviewNet	87.5%	43.75%	81.25%
AA-MultiviewNet with Self Attention	100%	37.5%	87.5%

Overall, the self-attention mechanism provided modest improvements in cross-subject generalization, increasing overall test accuracy by 4.17%. However, the enhancement was inconsistent across patients and tasks, with some individuals showing substantial gains while others experienced performance decreases. The persistent train-test gap and the continued occurrence of complete classification failures indicate that overfitting remains a significant challenge despite architectural modification.

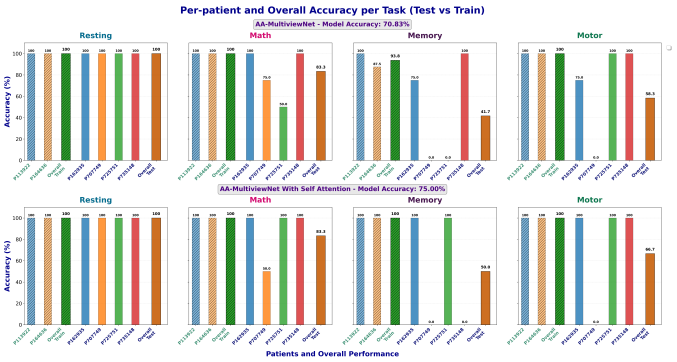

Fig. 4: Improved model performance on Cross dataset.

6. Discussion

Despite employing a sophisticated deep learning architecture capable of extracting both spatial and temporal features from MEG signals, our model struggled to generalize in cross-subject classification. Even after incorporating self-attention mechanisms designed to capture long-range spatial dependencies, performance remained suboptimal, with only a modest accuracy gain from 70.83% to 75.00%. This indicates that architectural improvements alone are insufficient to overcome the deeper limitations imposed by the dataset.

The poor generalization observed in our cross-subject experiments can be primarily attributed to insufficient training data diversity. To support this claim, we performed Principal Component Analysis (PCA) with k-means clustering (K = 3), as shown in Fig. 5, which revealed a stark separation between training and test samples across different cognitive tasks. Notably, in Memory and Motor tasks, test samples formed entirely separate clusters from training samples, indicating that test patients exhibited substantially different neural activity patterns compared to training patients. This separation demonstrates that the test data distribution falls largely outside the learned feature space, creating a fundamental mismatch that challenges the model's ability to generalize effectively.

To further investigate the effect of data availability, we conducted an additional experiment in which the cross-subject model with self-attention was trained using both intra and cross-subject train data. Under identical hyperparameter settings, this combined training led to a notable improvement in overall test accuracy, increasing it to 82%. This finding supports the hypothesis that the core bottleneck is not model design, but rather insufficient and non-representative training data.

Conversely, the excellent intra-subject performance achieved by our model (100% accuracy) validates the effectiveness of the chosen architecture in capturing task-specific neural patterns when sufficient participant-specific data is available. This demonstrates that the model is indeed capable of learning meaningful task-related brain signal

features within an appropriate data regime where training and testing distributions remain consistent.

Taken together, these findings underscore the critical importance of dataset diversity and representativeness in MEG classification tasks. While sophisticated model architectures are necessary, they must be supported by sufficient training examples that adequately span the variability inherent in real-world neuroimaging data. The observed performance gap between intra- and cross-subject classification highlights a fundamental challenge in neuroimaging: the substantial inter-individual variability in neural responses that current datasets may not adequately capture.

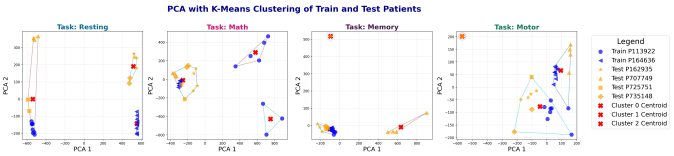

Fig. 5: PCA with K-Means Clustering of Train and Test Patients. Principal Component Analysis (PCA) visualization showing the first two principal components (PC1 and PC2) for patient data across the four different cognitive tasks. Blue points represent training patients, while yellow points represent test patients. Different shapes (circles, triangles, diamonds, squares) correspond to individual patient IDs as indicated in the legend. K-means clustering (K = 3) identified three distinct clusters with centroids marked by red X symbols. The analysis reveals important differences in data distribution between training and test sets across tasks. While some tasks (*resting* and *math*) show mixed clusters containing both training and test patients, other tasks (*memory* and *motor*) exhibit clusters composed primarily or exclusively of test samples. This separation suggests that test patients in the latter tasks have substantially different neural activity patterns compared to training patients, which could negatively impact the model's ability to generalize effectively to the test set.

7. Conclusion

In this study, we presented a deep learning framework tailored to the classification of cognitive states present in MEG data. The model is based on a multiview architecture with a spatial and a temporal branch. While the model achieves perfect performance in both training and testing for intra-subject classification, it struggles greatly to generalize across multiple subjects, especially those subjects that the model had not seen before in training. To remedy this, we introduced self-attention into the model with varying success. Several test sets saw marginal to major improvements in their accuracy, while one test set actually saw a decrease in accuracy when self-attention was used. Overall, self-attention improves the overall accuracy and stability in cross-subject settings. It does, however, not completely overcome cross-subject variability, suggesting that it can overtrain on noise present in the data.

In addition, the observed limitations in cross-subject generalization highlight the complex and highly individualized nature of brain activity patterns. This suggests that a one-size-fits-all approach may be insufficient for real-world deployment. The incorporation of subject-specific calibration methods, or transfer learning strategies could help bridge the gap between personalized and generalized performances, allowing the model to perform better on previously unseen data.

These results illustrate the need for more robust feature extraction and generalization methods in brain signal classification. For future research, there should be a focus on expanding training datasets, exploring more adaptive learning approaches, including more computational power to research better fitting parameters, such as a bigger window sizes, and integrating model interpretability to support clinical approaches.

References

[1] Muhammad Imran Khalid et al. "MEG data classification for healthy and epileptic subjects using linear discriminant analysis". In: *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. 2015, pp. 360–363. DOI: 10.1109/ISSPIT.2015.7394360.

[2] Manuel Lopez-Martin, Angel Nevado, and Belen Carro. "Detection of early stages of Alzheimer's disease based on MEG activity with a randomized convolutional neural network". In: *Artificial Intelligence in Medicine* 107 (2020), p. 101924. ISSN: 0933-3657. DOI: https://doi.org/10.1016/j.artmed.2020.101924. URL: https://www.sciencedirect.com/science/article/pii/S0933365720300749.

[3] Li Zheng et al. "EMS-Net: A Deep Learning Method for Autodetecting Epileptic Magnetoencephalography Spikes". In: *IEEE Transactions on Medical Imaging* 39.6 (June 2020), pp. 1833–1844. ISSN: 1558-254X. DOI: 10.1109/TMI.2019.2958699. URL: https://ieeexplore.ieee.org/abstract/document/8930587.

[4] Ismail Alaoui Abdellaoui et al. "Deep brain state classification of MEG data". In: *arXiv preprint arXiv:2007.00897* (2020).

[5] Delshad Vaghari, Ehsanollah Kabir, and Richard N Henson. "Late combination shows that MEG adds to MRI in classifying MCI versus controls". In: *Neuroimage* 252 (2022), p. 119054.

[6] Brian Kenji Iwana and Seiichi Uchida. "An empirical survey of data augmentation for time series classification with neural networks". In: *Plos one* 16.7 (2021), e0254841.

[7] Szilvia Szeghalmy and Attila Fazekas. "A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning". In: *Sensors* 23.4 (2023), p. 2333.

[8] Takuya Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. 2019, pp. 2623–2631. DOI: 10.1145/3292500.3330701.