

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics



Neuro-Computing Methods for Major Depressive Disorder Detection and Psychotherapy Aid

Disertation Thesis

Cheng Kang

Ph.D. programme: Bioengineering
Supervisor: Doc. Ing. Daniel Novak, Ph.D.

Prague, February 2025

Thesis Supervisor:

Doc. Ing. Daniel Novak, Ph.D.
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2
160 00 Prague 6
Czech Republic

Declaration

I hereby declare I have written this doctoral thesis independently and quoted all the sources of information used in accordance with methodological instructions on ethical principles for writing an academic thesis. Moreover, I state that this thesis has neither been submitted nor accepted for any other degree. The results presented in this dissertation have been published in [1]–[4] during my Ph.D. research in cooperation with my dissertation supervisor Daniel Novak. In my Ph.D. study, I collaborated with several researchers on multiple projects. I publish articles with Yong Hu [1], [2], [5]–[7], Yuezhi Li [2], [5], [6], Huiyu Zhou [3], [7], Yudong Zhang [2], [8]–[10], Xujing Yao [10]–[12], Jindrich Prokop [7], Xiang Yu [8], [9], Xinye Chen [13], [14] and Erin Carson [13], [14].

In Prague, February 2025

.....
Cheng Kang

Abstract

The detection of Major Depressive Disorder (MDD) has made significant strides through the integration of neurocomputing techniques and traditional machine learning methods. Additionally, digital-based psychological therapy approaches have proven to be reliable and convenient in aiding depression treatment. However, despite the advancements, detection rates for depression remain insufficient for consistent clinical application. Furthermore, digital psychological therapy approaches are often limited by location and scheduling constraints, reducing their accessibility and effectiveness. To address these challenges, this thesis proposes two noninvasive methods for detecting MDD using electroencephalogram (EEG) signals, providing clear visualization results and stable accuracy. Additionally, to enhance the effectiveness of psychotherapy for MDD patients, this thesis develops a fine-tuning method for language models and releases a psychotherapy-focused digital dataset. Lastly, this work introduces a method to integrate the semantic representation of EEG signals into natural language processing.

In Chapters 2 and 3, I have designed a noninvasive system to visualize dynamic functional brain networks in both depressive patients and healthy controls during Working Memory (WM) tasks. Two residual neural networks (ResNets), trained on selected EEG channels and frequencies, effectively detect depression and assess its severity. Indirectional and directional brain functional dynamics highlight the differences between depressive patients and healthy controls, providing reliable interpretability of artificial neural network (ANN) models.

In Chapters 4 and 5, I have proposed a neuroscience-inspired architectural model incorporating shunting inhibition to develop advanced training and fine-tuning methods for pre-trained language models. I also have presented a psychotherapy dataset optimized by Large Language Models (LLMs). The proposed fine-tuning method allows for gating tunable weights on downstream language tasks, while the psychotherapy dataset enables LLMs to access professional and widely-accepted therapeutic knowledge.

In Chapter 6, I have introduced a method for enabling LLMs to understand time-series data by converting it into symbolic series. This tool equips LLMs with an internal symbolic chain-of-pattern for more effective processing of time-series information.

Collectively, these components form a closed-loop system for depression detection and psychotherapy support, seamlessly merging EEG signals and natural language processing to deliver systematic, interactive, and visually interpretable results. Beyond contributing scientific insights for each individual system, this work establishes a practical framework for advancing future efforts in depression detection and psychotherapy enhancement.

Keywords: Depression detection, depressive severity scoring, brain computer interface, parameter efficient fine tuning, large language models, assistant instruction, psychotherapy chatbot, multi-modality of LLMs, time series.

Acknowledgements

I would like to express my heartfelt gratitude to all those who have contributed to the completion of my PhD thesis. This journey has been challenging, yet immensely rewarding, and I couldn't have reached this milestone without the support, encouragement, and assistance of numerous individuals.

- First and foremost, I am deeply thankful to my advisor, Daniel Novak, whose unwavering guidance, wisdom, and mentorship have been invaluable throughout my doctoral studies. Your dedication to my academic and personal growth has been instrumental in shaping the researcher I have become.
- Tomáš Sieger, Eduard Bakštein, Jiří Anýž, Jakub Schneider, Xinye Chen, Erin Carson and fellow PhD student, for joining discussion and helping me with some tedious task.
- Jindřich Prokop, Ihor Varha and Václav Burda, Xujing Yao, Lei Tong, Xiang Yu, these PhD students, for being always available to listen and share their opinion, not minding the time I have stolen from them.
- I would like to acknowledge the faculties and staffs in University of Hong Kong, University of Leicester and Shenzhen University, whose commitment to academic excellence provided a nurturing environment for my research endeavors. The resources, facilities, and intellectual stimulation provided by the university were indispensable. Yuezhi Li, Yong Hu, Huiyu Zhou and Yudong Zhang professors of computer-science and neuroscience, for answering a lot of rather naive questions and letting me work in such an interesting field by providing computer science knowledge on medical domain and by steering the team towards relevant problems.
- Lastly, I dedicate this work to Yuqing Chen, Patrik Jankuv and Fabián Bodnar, Štěpán Bořek, for helping me gather necessary research data.
- I am grateful to my family, my father Linchao Kang, my mother Xiuge Lei, my wife Qingyun Yang and my two little angels, for their unwavering love and encouragement. Your belief in me sustained my determination during the highs and lows of this academic pursuit.

Thank you all for being an integral part of this academic journey. Your contributions, whether big or small, have left an indelible mark on my life.

Contents

Abstract	iv
Acknowledgements	v
List of Tables	x
List of Figures	xii
1 Introduction	1
1.1 Goals of the Thesis	2
1.2 Challenges	3
1.2.1 Detecting Depression	4
1.2.2 Psychotherapy Using Large Language Models	6
1.3 Thesis Outline	10
2 Classifying and Scoring Depressive Disorders	13
2.1 Introduction	13
2.2 Related Works	15
2.2.1 Brain Regions and Functional Network Extraction	16
2.2.2 Utilization of Artificial Neural Networks	17
2.3 Methodology	17
2.3.1 EEG Recording and Participants	17
2.3.2 Working Memory Experiments	18
2.3.3 EEG Preprocessing	18
2.3.4 Residual Neural Networks	18
2.4 Result	19
2.4.1 Memory load comparison of behavioural results	19
2.4.2 The Connections comparison	19
2.4.3 Clusters between these Three Groups	20
2.4.4 Results of Classifying and Scoring MDD Patients	21
2.5 Discussion	22
2.5.1 Potential Inducing Factors for Depression	23
2.5.2 Topological Analysis	23
2.5.3 Contribution of Frequency and Topological Selection for Classifying and Scoring Depressive Patients	24
2.5.4 State of the Art for Classifying Depressive Patients	25
2.5.5 State of the art for scoring depressive severities	25
2.6 Conclusion and future work	25

3	Noninvasive Visualization of Brain Networks	29
3.1	Introduction	30
3.2	Related Work	31
3.2.1	Pathway for Attention Arousal and Executive Function	31
3.2.2	Pathway for Coding and Decoding	31
3.2.3	Pathway for Sustained Brain Activity	31
3.2.4	Pathway for Lateral Inhibition	32
3.3	Methods	32
3.3.1	Participants	32
3.3.2	Experimental Procedures	32
3.3.3	EEG Recording	33
3.3.4	Data Analysis	34
3.4	Study Results	37
3.4.1	Behavioral Results	37
3.4.2	Scalp Topography Performance	37
3.4.3	Band-Specific Synchrony Analysis	38
3.4.4	Band-Specific Directionality Analysis	38
3.4.5	Neurocognitive Architecture and Component Processes of Working Memory	39
3.5	Discussion	39
3.5.1	The Maintenance Loop During WM	40
3.5.2	The Inhibition Loop During WM	41
3.5.3	Conclusion and Future Directions	41
4	Inhibition Adaption On Pre-trained LMs	47
4.1	Introduction	47
4.2	Problem Statement	49
4.3	Explanation of Shunting Inhibition	50
4.3.1	Shunting Inhibition (Gate with Inhibition)	50
4.3.2	Membrane Potentials and Threshold	51
4.4	Related Work	51
4.4.1	Transformer-based Language Models	51
4.4.2	Fine-tuning on NLP Downstream Tasks	52
4.4.3	Parameter-Efficient Fine-Tuning	52
4.4.4	Threshold and Inhibition	53
4.5	Inhibition Adaptation	54
4.5.1	Inhibited Adaptation	54
4.5.2	Inserting InA into Transformer	55
4.6	Experiments	56
4.6.1	Experiment Settings	56
4.6.2	Evaluation Datasets	57
4.6.3	Fine-Tuning Implementation Details	57
4.7	Results	57
4.7.1	Efficiency: Trainable Parameters and Speed	58
4.7.2	Effectiveness: InA on Fine-tuning	58
4.7.3	InA on the Text Classification Task	59
4.8	Analysis and Discussion	63
4.8.1	Difference Between LoRA and InA	63

4.8.2	Should We Need Inhibition During Fine-Tuning? How Does It Work?	64
4.8.3	How to Choose the Inhibition Level Inh_p and Select a Good Rank r in Real Cases?	65
4.8.4	Can InA Really Inhibit Irrelevant Knowledge? How Can It Do So?	65
4.9	Conclusion	66
5	Domain Specific Assistant Instruction For LLMs	76
5.1	Introduction	76
5.2	Problem Statement	78
5.3	Related Work	79
5.3.1	Psychotherapy-based Conversational Systems	79
5.3.2	Instruction Data for Language Models	79
5.3.3	Parameter-Efficient Fine-Tuning of Pre-trained Language Models	80
5.4	Methodology	81
5.4.1	Data Collection	81
5.4.2	Assistant on Annotation and Task Identification	81
5.4.3	Assistant on Generation, and Evaluation	82
5.5	Experiments	83
5.5.1	Experiments Settings	83
5.5.2	Models	84
5.5.3	Metrics	84
5.5.4	Analytic Experiments	85
5.6	Conclusion	88
6	LLM-ABBA For Digital Health	89
6.1	Introduction	89
6.2	Related work	93
6.3	Methodologies	94
6.3.1	ABBA Symbolic Approximation	94
6.3.2	Error Analysis Reconstruction	97
6.3.3	ABBA to LLM	99
6.3.4	Linguistic Investigation: Zipf's Law	101
6.4	Experiments	102
6.4.1	Hyperparameters	103
6.4.2	Compression and Recovery	104
6.4.3	Time Series Classification Tasks	105
6.4.4	Time Series Regression Tasks	106
6.4.5	Time Series Forecasting Tasks	107
6.4.6	QLoRA Fine-Tuning	107
6.4.7	Semantic consistency	108
6.5	Limitations	108
6.6	Conclusion	109
7	Conclusion	112
7.1	Summary	112
7.2	Contributions and Achievements	113
7.3	Future Work	114

List of Candidate’s Publications Related to the Thesis **116**

7.4 Publications in Impacted Journals 116

7.5 Other Publications 117

References **139**

List of Tables

2.1	The comparison of reaction time and response accuracy rates between two different memory loads (average \pm standard deviation) in two depressive groups.	20
2.2	Classification (Accuracy) and scoring depression (RMSE) results (mean and standard deviation) using whole frequency bands (delta, theta, alpha and beta).	21
2.3	Classification (Accuracy) and scoring depression (RMSE) results (mean and standard deviation) using beta frequency bands.	21
2.4	Classification (Accuracy) and scoring depression (RMSE) results (mean and standard deviation) using whole frequency bands (delta, theta, alpha and beta) and selected EEG channels.	22
2.5	By scaling the size of proposed ResNets, the below shows the classification (Accuracy) and scoring (RMSE) results using beta frequency band and selected EEG channels.	22
2.6	Comparison with existing methods on classifying depression with EEGs. .	25
2.7	Comparison with existing methods on scoring depressive severities with EEGs.	26
4.1	Hyper-parameters for fine-tuning BERT, RoBERTa and DeBERTa with inhibited gate MLPs mechanism on down-streaming tasks.	56
4.2	The efficiency of InA and other adaptation FT methods in terms of trainable parameters, inference (complexity), and update speed (back-propagation). .	58
4.3	Comparison results of fine-tuning the GLUE development set on <i>BERT – large</i> , <i>RoBERTa – large</i> , <i>DeBERTaV2 – large</i> and <i>DeBERTaV3 – large</i> with <i>InA</i> (inhibition level percentile is 0.3). † indicates runs configured in a setup similar to [46] for a fair comparison.	59
4.4	Comparison results of fine-tuning SQuAD v1.1, SQuAD v2.0 and SWAG on <i>BERT – large</i> , <i>RoBERTa – large</i> , <i>DeBERTaV2 – large</i> and <i>DeBERTaV3 – large</i> with <i>InA</i> (inhibition level percentile is 0.9). ★ indicates being run under the original configuration for a fair comparison. (Note that missing results in the literature are signified by ‘-’).	59
4.5	When using different activation functions, we set the inhibition level percentile at 0.3 and present the comparison results on the GLUE development set within five epochs fine-tuning based on <i>BERT – large</i>	61
4.6	Comparison results on fine-tuning the GLUE development set, SQuAD v1.1, SQuAD v2.0, and SWAG—Inserting InA into <i>BERT – large</i> (1*), <i>RoBERTa – large</i> (2*) and <i>DeBERTa – large</i> (3*). The values after each model are inhibition levels.	62

4.7	Comparison results on fine-tuning the GLUE development set, SQuAD v1.1, SQuAD v2.0, SWAG and NER. (Note that Key* and Query* respectively mean inserting InA into Transformers' Key side and Query side).	63
4.8	Comparison results on fine-tuning the GLUE development set, SQuAD v1.1, SQuAD v2.0, SWAG and NER on language models' several last layers.	64
5.1	Prompt used for identifying the type of tasks.	82
5.2	Prompt used for generation and evaluation.	83
5.3	Hyper-parameters for querying OpenAI API in different experiments. . . .	83
5.4	Hyper-parameters for fine-tuning pre-trained LLMs in different experiments.	84
5.5	The manually constructed Instruction and GPT-4 revised Assistant-Instruction on the Depressive Disorder domain.	85
5.6	Based on Llama2-7B, we illustrate the performance of Zero-Shot, inhibited LoRA Tuned and RAG methods on Psychotherapy data.	86
5.7	For evaluating the performance of LLM on psychotherapy domain, two methods - inhibited LoRA and RAG - were used on two pre-trained LLM have been tuned on Assistant-Instruction using	87
6.1	Hyperparameters of Classification tasks. Quant. is the model quantization process. Inhib. is the inhibition threshold in QLoRA. Embed. means to save tuned embeddings. Optimis. is the optimization method. LR is the learning rate. Acc. is the accyrcy rate (%).	103
6.2	Hyperparameters of Regression tasks. Quant. is the model quantization process. Inhib. is the inhibition threshold in QLoRA. Embed. means to save tuned embeddings. Optimis. is the optimization method. RMSE is the root-mean-square-error.	103
6.3	Hyperparameters of Prediction tasks. Quant. is the model quantization process. Inhib. is the inhibition threshold in QLoRA. Embed. means to save tuned embeddings. Optimis. is the optimization method. MAE is the mean-absolute-error, and MSE is the mean-square-error.	104
6.4	Symbolic approximation performance on ETTh1 data using ABBA. ABBA describes a time series sample by using symbolic approximation, and the number of used symbols depdnds on the complexity of the data. If the time series sample is a regular wave (for example, a sine wave), the number of used symbols is small; otherwise, ABBA needs more symbols.	105
6.5	Full comparison of results for time series classification tasks(%) on UCR datasets.	110
6.6	Full comparison of results on medical time series classification tasks(%) on EEG eye states, ptb-db, and MIT-BIH.	110
6.7	Full comparison of results on the regression task on 19 Monash Time Series Regression datasets.	111
6.8	Full comparison of results for the prediction task on 4 time series prediction datasets.	111
6.9	The performance of LLM-ABBA with extra new tokens (symbolic ASCII codes) on ETTh1 data in terms of time series forecasting tasks.	111

List of Figures

1.1	The whole framework of depression detection and psychotherapy assistance in this thesis.	3
1.2	The framework of depressive severity scoring system.	4
1.3	A semantic graph that describes how Assistant-Instruction can change the professional embedding to a common embedding. A successful model is expected to use the provided instructions (including task and domain definition examples) to response to professional evaluation.	7
1.4	The integration of time series and LLMs demonstrates potential in solving complex real-world problems.	9
2.1	The framework of depressive severity detecting and scoring system using EEG signals. The entire procedure about classifying depression and scoring depressive severity (A1 → A2 → A3 → A4).	15
2.2	The structure of the constructed residual neural network. The input size is $64 \times 64 \times 18$ or $16 \times 64 \times 18$. Conv+BN+ReLu means the processing of convolution (Conv), batch normalization (BN) and rectified linear unit (ReLU). FCL is the fully connected layer. The shortcut is purely forward plus. $\times 3$ means this block should be repeated triple times.	19
2.3	The number of the significant pairs in terms of the comparison between 2-back and 0-back tasks.	21
2.4	The t values (significant level) of the comparison between the depression group and the healthy control group.	27
2.5	Clustering of significantly increased and decreased phase synchronization indices primarily in the beta bands for both depressive groups and the control group. The upper panels (A and B) show the significant PSI decrease and increase during the 2-back task, compared to the 0-back condition ($p < 0.05$) between the depressive group with low scores and the control group. The lower panels (C and D) show the significant PSI decrease and increase between the depressive group with high scores and the control group. Clusters A, B, C, and D represent significant groupings identified with a family-wise error rate correction at $\alpha = 0.01$. The panels labeled Bc, Cc, Cd, and Dc show correlation coefficients for phase synchronization within the corresponding clusters. The gray panel (C) indicates that the significance level is slightly weaker.	28
3.1	Experimental procedure and timeline. Participants responded to stimuli by pressing the '1' key with the index finger for target stimuli (match) and the '2' key with the middle finger for nontarget stimuli (mismatch).	33

3.2	Scalp voltage maps for the 2-back condition minus the 0-back condition, showing distinct activation patterns in the front and back hemispheres during different time periods. The circled electrode sites correspond to Fz and Oz. The Global Field Power, which represents the sum of squared amplitudes across all channels, is shown in a logarithmic scale.	34
3.3	RSs and their corresponding time courses of group average EEGs. The left panel shows the three directional time courses of the RSs, and the right panel shows the locations and orientations of the four RSs, with orientation 1 representing the primary orientation of each RS.	35
3.4	Phase-locked connections among four sources from 0 ms to 700 ms (a, b) and from 700 ms to 1600 ms (c, d). (a) Left panel shows connections at specific frequencies, with the right panel displaying circular statistical angles and their distribution. Circular histograms also illustrate the mean angles of phase differences between pairs of sources (red line). (b, d) t-statistics for the differences in PLV between 2-back and 0-back tasks across subjects. For example, in the pair of S1 and S3, the PLV in the 18-21 Hz beta band was higher during the 2-back task, peaking at 20 Hz. The green band represents the t-values for a one-sample t-test with a 95% confidence interval using a bootstrap method, and the red line represents the t-value. (c) Connections at specific frequencies with their circular statistical angles and distribution.	43
3.5	Directed connections based on the time-varying GPDC. (a) Time-frequency representations of the time-varying GPDC under the 2-back task, with significant grey blocks indicating differences between the 0-back and 2-back tasks using a two-sample t-test. The bar represents the GPDC value. (b) Directed connections at different latencies, indicated by color-coded arrows representing the direction and strength of information flow. Early latency intervals (I: 150–300 ms, E; II: 550–700 ms, D) primarily involve S3→S4 (E) and S2→S3 (D), both reflecting trigger information transmission. Late latency intervals (III: 700–900 ms, A; IV: 900–1100 ms, C; V: 1300–1600 ms, B, G, H) show diverse connections between sources reflecting memory encoding, updating, and sustained attention processes.	44
3.6	Schematic explanation of representations to brain networks during WM tasks. Left upper panel is the location illustration of four fitted sources. A~E present components relative to WM in terms of some specific neurocognitive processes. A . During this duration, selective attention is activated by the trigger of capitals shown on the screen, and this induced the attention mechanism in PPC cortex. B . Executive and cognitive functions between right PFC cortex and left PPC region, appear after selective attention being implemented to process numerical and verbal information. C . The PFC and right hemisphere connections indicate the update of information flow for memory storing, and lateral inhibition to avoid the failure of memory representation. D . Persistence of information under WM tasks happens in PFC cortex. E . The last process for the recall of sustained attention, lateral inhibition to avoid the failure of attention and memory processing, as well as disinhibition.	45

3.7	Summary of the proposed neurocognitive architecture for WM. X represents the visual n-back task trigger. Before responses, attention arousal (0-1-2) is linked with the activity maintenance loop (2-3-2). After the response, the brain enters a memory maintenance loop, consisting of an activity loop (2-3-5-2) and a major memory loop (3-5-3), alongside inhibition or disinhibition loops (2-3-4-2 , 2-3-5-4-2). The central role of inhibition is crucial for maintaining accuracy in information processing, while disinhibition resets brain activity, enabling subsequent cognitive processes.	46
4.1	Illustration of the transformer architecture and our proposed parameter-efficient tuning method: Inhibition Adaptation.	49
4.2	A practical example of InA and its use in the $BERT_{large}$ model, which has been fine-tuned under question-answering datasets.	50
4.3	Inspiration from Neuroscience: Gate with Inhibition.	51
4.4	Plots of corresponding metrics according to the number of epochs on the validation split of GLUE, SQuAD v1.1, SQuAD v2.0 and SWAG. The giBERT means inserting InA (gate inhibition mechanism) into BERT. . . .	60
4.5	Roughly disassembled DeBERTaV3 architecture.	63
4.6	From left to right, fine-tuning $BERT - large$ on CoLA with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), e) InA(0.9).	68
4.7	From left to right, fine-tuning $BERT - large$ on SQuAD with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), e) InA(0.9).	69
4.8	From left to right, fine-tuning $RoBERTa - large$ on SQuAD-V2 with no-InA, InA($In_p = 0.0$), InA($In_p = 0.1$), InA($In_p = 0.3$), InA($In_p = 0.9$). . . .	70
4.9	From left to right, fine-tuning $Llama2$ on SQuAD-V2 with no-InA, InA($In_p = 0.0$), InA($In_p = 0.1$), InA($In_p = 0.3$), InA($In_p = 0.9$).	71
4.10	From left to right, fine-tuning $BERT - large$ on RTE with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), e) InA(0.9).	72
4.11	From left to right, fine-tuning $BERT - large$ on MRPC with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), e) InA(0.9).	73
4.12	From left to right, fine-tuning $BERT - large$ on QNLI with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), e) InA(0.9).	74
4.13	From left to right, fine-tuning $BERT - large$ on SWAG with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), e) InA(0.9).	75
5.1	Schematic representation of Assistant-Instructional prompts in psychotherapy domains. Step one: Data reformatting; Step two: Task identification; Step three: Knowledge expansion; Step four: Evaluation.	77
5.2	Schematic representation of model fine-tuning and the interaction between Chatbot and User.	80
6.1	The integration of time series and LLM demonstrates potential in solving complex real-world problems.	90

6.2	Plot (a) shows a sine function with 1,000 points, and (b) shows the ECG-FiveDays time series from the UCR Archive. We first perform fABBA with $\text{tol} = 0.1$ and $\alpha = 0.1$ and perform SAX with approximately the same length of symbolic representation and the number of distinct symbols. In plot (a), fABBA generates symbols “aBbCbCbCbCbCbCbCA” while SAX generates symbols “aACBbaACBbaACBbaAABb”; in figure (b), fABBA generates symbols “fAcaDECeBdbF” while SAX generates symbols “AAAAAABbCcDaaaAaa”.	92
6.3	The LLM-ABBA framework: Given an input time series, we first transform and compress the time series into a symbolic series via steps ① and ①. These symbolic series are then tokenized using the LLM’s tokenizer ②. The instruction containing the symbolic series is also tokenized by the LLM’s tokenizer ②. By fine-tuning the pretrained LLM, the QLoRA with inhibition mechanism is applied in both ③ and ③. To implement the corresponding tasks, ④ and ⑤ load the LLM according to the task type, while ④ loads the LLM for generation tasks. For symbolic series inversion, ⑥ and ⑤ use ABBA to decompress the generated symbolic series. Finally, in ⑦ and ⑥, the output time series from LLM-ABBA is projected to generate forecasts.	99
6.4	A synthetic trigonometric sine series with 1,000 points is generated, and symbolic approximation using 4 symbols is separately performed with APCA (the upper panel) and FAPCA (the lower panel) on the time series.	100
6.5	Frequency and rank of symbols in various UCR datasets.	102
6.6	Visualization of reconstructed input-168-predict-24 results on ETTh1 data by using ABBA symbolic approximation, where $\text{tol} = 0.01$, $\alpha = 0.01$ and $\text{scl} = 3$	104
6.7	Visualization of input-168-predict-24 results on ETTh1 using LLM-ABBA.	107

List of Acronyms

- ABBA** Adaptive Brownian Bridge-based symbolic Aggregation. [3](#), [89](#), [91–93](#), [95–97](#), [101–109](#), [113](#), [114](#)
- ADHD** Attention Deficit-Hyperactivity Disorder. [30](#)
- AI** Artificial Intelligence. [4](#), [79](#)
- ALBERT** A Lite BERT. [51](#)
- ANN** Artificial Neural Network. [2](#), [17](#), [26](#), [50](#), [112](#)
- APCA** Adaptive Piecewise linear Continuous Approximation. [95](#), [99](#), [100](#)
- BDI** Beck Depression Inventory. [4](#)
- BERT** Bidirectional Encoder Representations from Transformers. [48](#), [51](#), [52](#), [54](#), [57](#), [58](#), [61](#), [62](#), [102](#)
- BiRNN** Bidirectional Recurrent Neural Networks. [106](#)
- BOLD** Blood-Oxygen-Level-Dependent. [40](#)
- CBT** Cognitive Behavioral Therapy. [1](#), [14](#)
- CNN** Convolution Neural Network. [17](#), [18](#), [106](#)
- CoLA** Corpus of Linguistic Acceptability. [59](#), [61](#), [66](#)
- COP** Chain-Of-Pattern. [91–93](#), [103](#), [106](#), [108](#)
- DeBERTa** Decoding-enhanced BERT with Disentangled Attention. [48](#), [51](#), [52](#), [57](#), [58](#), [61](#), [62](#), [65](#)
- DLPFC** Dorsolateral Prefrontal Cortex. [24](#), [40](#)
- DTW** Dynamic Time Warping. [108](#)
- ECG** electrocardiogram. [102](#), [106](#)
- EEG** electroencephalogram. [2–6](#), [10](#), [11](#), [13–18](#), [23](#), [24](#), [26](#), [29](#), [30](#), [32–35](#), [102](#), [105](#), [106](#), [108](#), [112–114](#)
- EHR** Electronic Health Record. [1](#), [2](#), [114](#)

- ELU** Exponential Linear Unit. [49](#)
- EOG** Electrooculogram. [33](#)
- ERP** Evoked Related Potential. [34](#), [35](#)
- fABBA** Fast Adaptive Brownian Bridge-based symbolic Aggregation. [91](#), [92](#)
- FAPCA** Fixed-point Adaptive Piecewise linear Continuous Approximation. [99](#), [100](#), [109](#)
- FDR** False Discovery Rate. [37](#)
- FFN** Feed-Forward Network. [52](#)
- fMRI** Magnetic Resonance Imaging. [4–6](#), [17](#), [31](#), [35](#), [39](#), [40](#)
- fNIRS** Functional near-infrared spectroscopy. [4](#), [6](#), [31](#)
- FPT** Frozen Pretrained Transformer. [93](#)
- FT** fine-tuning. [57–60](#)
- GELU** Gaussian Error Linear Unit. [55](#)
- GeLU** Gaussian Error Linear Unit. [49](#), [58](#), [61](#)
- GLoRA** Generalized LoRA. [53](#), [80](#)
- GLUE** General Language Understanding Evaluation. [57](#), [59](#), [61](#), [62](#), [65](#)
- gMLP** gate multilayer perceptron. [64](#)
- GPDC** General Partial Directed Coherence. [11](#), [32](#), [34](#), [36–39](#)
- GPT** Generative Pre-trained Transformer. [54](#), [58](#), [102](#)
- HAMD** Hamilton Depression Rating Scale. [4](#), [18](#), [19](#)
- HHV-6** Human Herpesvirus 6. [23](#), [26](#)
- InA** Inhibition Adaption. [11](#), [47–50](#), [54](#), [55](#), [57–67](#), [80](#), [83](#), [113](#)
- KNN** K-Nearest Neighbor. [16](#)
- L-DLPFC** Left-Dorsolateral Prefrontal Cortex. [24](#)
- LeakyReLU** Leaky Rectified Linear Unit. [49](#), [58](#), [61](#)
- LLM** Large Language Model. [2–4](#), [7–11](#), [76–78](#), [82–84](#), [86–93](#), [96](#), [101–109](#), [112–114](#)
- LLM-ABBA** Large Language Model with Adaptive Brownian Bridge-based symbolic Aggregation. [11](#)
- LM** Language Model. [6](#), [7](#), [11](#), [47–49](#), [52](#), [54](#), [55](#), [60](#), [64](#), [79](#)

- LoRA** Low-Rank Adaption. 6, 7, 47–49, 53, 58, 63, 64, 80, 86
- LSTM** Long Short Term Memory. 17, 106
- MAE** Mean Absolute Error. 105, 107
- MDD** Major Depressive Disorder. 1, 4, 5, 13, 14, 18, 19, 24
- MLP** Multilayer Perception. 3, 57, 64, 112
- MRI** Magnetic Resonance Imaging. 25
- MRPC** Microsoft Research Paraphrase Corpus. 59–61, 66
- MSE** Mean-Square Error. 105, 107
- MVAR** Multivariate Autoregressive. 36
- NLP** Natural Language Processing. 1, 6, 47, 51, 52, 77
- NLU** Natural Language Understanding. 7, 48, 57, 59, 64
- PDC** Partial Directed Coherence. 32
- PEFT** Parameter-Efficient Tuning Method. 80, 81
- PET** Positron Emission Tomography. 31
- PFC** Prefrontal Cortex. 11, 29–31, 38–41, 112
- PLC** Phase Lock Coherence. 11, 32, 34
- PLSR** Partial Least Squares Regression. 25
- PLV** Phase Lock Value. 35, 36, 39
- PPC** Posterior Parietal Cortex. 29–31, 38–41, 112
- PSI** Phase Synchrony Index. 14–16, 19, 20, 22
- QLoRA** Quantized LoRA. 53, 92, 102–104, 106, 107
- QNLI** Stanford Question Answering Dataset. 61, 66
- QQP** Quora Question Pairs. 59, 61
- RAG** Retrieval-Augmented Generation. 76, 86
- ReLU** Rectified Linear Unit. 49, 58, 61
- ResNet** Residual Neural Network. 13, 14, 16, 17, 19, 24–26
- RF** Random Forest. 17
- RLHF** Reinforcement Learning on Human Feedback. 8, 76

- RMSE** Root-Mean-Square Error. [13](#), [22](#), [25](#), [106](#)
- RoBERTa** Robustly Optimized BERT. [48](#), [51](#), [52](#), [57](#), [61](#), [64](#), [65](#)
- RS** Regional Source. [35](#), [36](#)
- RTE** Recognizing Textual Entailment. [60–62](#), [65](#), [66](#)
- RVR** Relevance Vector Regression. [25](#)
- SAX** Symbolic Aggregate approXimation. [91](#), [92](#), [108](#)
- SCID-CV** DSM-IV Axis I Disorders, Clinician Version. [4](#), [18](#), [19](#)
- SELU** Scaled Exponential Linear Unit. [49](#), [61](#)
- SNN** Spike Neural Network. [53](#), [54](#)
- Softmax** Softmax. [58](#)
- SOTA** State-Of-The-Art. [11](#), [51](#), [52](#), [76](#), [80](#), [89](#), [93](#), [103](#), [106–109](#), [112](#)
- SQuAD** Stanford Question Answering Dataset. [57](#), [60](#), [62](#), [65](#)
- SSE** Sum of Squared Errors. [96](#)
- SST2** Stanford Sentiment Treebank. [61](#)
- STSA** Symbolic Time Series Approximation. [89](#), [91](#), [93](#), [99](#), [108](#)
- SVM** Support Vector Machine. [17](#)
- SWAG** Situations With Adversarial Generations. [57](#), [60](#), [61](#), [65](#), [66](#)
- TSER** Time Series Extrinsic Regression. [89](#), [106](#)
- WHO** World Health Organization. [1](#)
- WM** Working Memory. [17](#), [19](#), [20](#), [22–24](#), [29–32](#), [39–41](#), [112](#), [113](#)

Chapter 1

Introduction

Depression, also known as [Major Depressive Disorder \(MDD\)](#), is a prevalent mental health condition globally. According to the [World Health Organization \(WHO\)](#), it affects approximately 3.8% of the global population, with a higher prevalence of 5.0% among adults and 5.7% among individuals aged 60 and older [15]. It is estimated that around 280 million people worldwide suffer from depression [15]. Unlike typical mood fluctuations or brief emotional responses to daily life events, depression becomes particularly concerning when it is recurrent and of moderate to severe intensity, often leading to significant impairments in daily functioning, including work, school, and family life. In the most severe cases, depression can result in suicide, which accounts for over 700,000 deaths annually. Suicide is the fourth leading cause of death among individuals aged 15-29 years.

Despite the availability of effective treatments, over 75% of individuals in low- and middle-income countries do not receive adequate care for mental health disorders [16]. Barriers to effective treatment include limited resources, a shortage of trained healthcare providers, and the social stigma surrounding mental illness. Moreover, depression is often misdiagnosed, with individuals who do not have the disorder sometimes wrongly diagnosed and prescribed antidepressants unnecessarily. Misdiagnosis can lead to a range of complications, including self-medication, substance abuse, inappropriate treatment, social isolation, and reduced performance in educational or professional settings [17]–[19]. For mild to moderate depression, [Cognitive Behavioral Therapy \(CBT\)](#) is generally considered the most effective treatment. For more severe forms of depression, a combination of psychotherapy and antidepressant medications is the current standard of care [20]–[22]. However, inadequate or delayed treatment can result in relapse and prolonged withdrawal symptoms [23].

[Natural Language Processing \(NLP\)](#) techniques have become integral in processing clinical notes and narratives, playing a critical role in handling [Electronic Health Record \(EHR\)](#) data [24]. One key application is improving clinical documentation by extracting

meaningful information from both structured data (e.g., lab results, vital signs) and unstructured data (e.g., clinical notes). Techniques like named entity recognition [25] and text summarization [26] are commonly used to identify relevant health-related entities and summarize treatment outcomes. Another significant application focuses on the analysis of [EHR](#) data to investigate disease progression [27] and adverse drug reactions [28]. These studies contribute to a deeper understanding of medical conditions and the effectiveness of therapeutic interventions. One more [EHR](#) application, such as EHRAgent, [29] enables autonomous code generation and execution to facilitate clinicians in directly interacting with [EHRs](#) using natural language.

This thesis explores how technology can enhance depression detection and support psychotherapy. Specifically, it focuses on leveraging [electroencephalogram \(EEG\)](#) signals for depression detection and employing [Large Language Models \(LLMs\)](#) to assist in therapeutic interventions. The aim of this thesis is to make these tools more accessible and effective for both patients and clinicians, improving the overall efficiency of diagnosis and treatment.

1.1 Goals of the Thesis

As illustrated in Figure [1.1](#), this thesis has two primary objectives:

1. **Noninvasive Depression Detection:** To develop a noninvasive system that uses [EEG](#) signals to assess the severity of depression. This involves analyzing functional brain networks to decode brain activity, facilitating depression diagnosis and providing insights into its severity.
2. **Psychotherapy Assistance:** To provide psychotherapy support through chatbots powered by [LLMs](#). These chatbots are designed to offer advice and guidance based on professional psychotherapy knowledge, supporting patients in their treatment journey.

In the aspect of depression detection, by integrating clinical knowledge of depression detection with advanced [Artificial Neural Networks \(ANNs\)](#), this system first demonstrates effective performance in both detecting depression and assessing its severity (as detailed in Chapter 2). However, visualizing brain abnormalities in depression patients alone does not yield sufficient insights, as we still do not know which step is abnormal, in terms of the information processing flow in the brain. The directional flow of brain networks provides a way to measure the information processing flow of the human brain. Thus, Chapter 3 introduces a noninvasive visualization method for mapping directional brain networks during working memory tasks using [EEG](#) signals. Additionally, there are

two works that contribute to the psychotherapy aids. To further enhance the performance of pretrained language models on specific downstream tasks, Chapter 4 presents a novel fine-tuning approach—Inhibited Gate [Multilayer Perceptions \(MLPs\)](#), inspired by the shunting inhibition mechanism. Moreover, Chapter 5 explores how psychotherapy data, refined and augmented by GPT-4, can be used to train other [LLMs](#) to generate effective and reliable therapeutic aids. Besides, because that [LLMs](#) cannot understand and analyze time series very well, especially [EEG](#) signals, Chapter 6 focuses on developing multimodal capabilities in [LLMs](#), specifically enabling them to process time-series data via [Adaptive Brownian Bridge-based symbolic Aggregation \(ABBA\)](#), thereby bridging the gap between [EEGs](#) signals and [LLMs](#). We finally make a close-loop that can fuse EEG signals and natural languages together, and this system also can provide with a systematic result, visually and interactively.

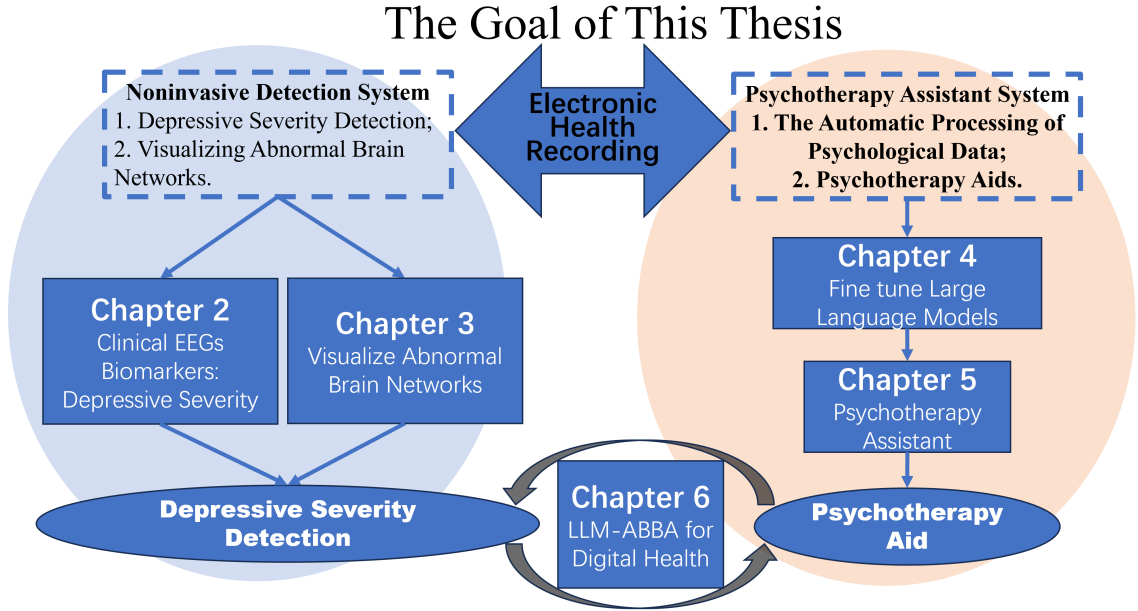


Figure 1.1: The whole framework of depression detection and psychotherapy assistance in this thesis.

1.2 Challenges

One of the challenges in treating depression is accurately diagnosing the condition and assessing its severity. Currently, doctors use brain scans, such as [EEGs](#), to measure brain activity and identify patterns indicative of depression. In this thesis, we employ a technique that combines [EEGs](#) data with machine learning to detect and assess the severity of depression. Another challenge is that psychotherapy, while essential for treating depression, is often difficult to access due to the need for trained therapists. By leveraging

LLMs—advanced [Artificial Intelligence \(AI\)](#) models trained to understand and generate human-like language—we aim to develop chatbots capable of providing basic psychotherapy support. This could make therapy more accessible and reduce the burden on human therapists.

1.2.1 Detecting Depression

Challenges of Detecting Depression and Scoring Depressive Severity

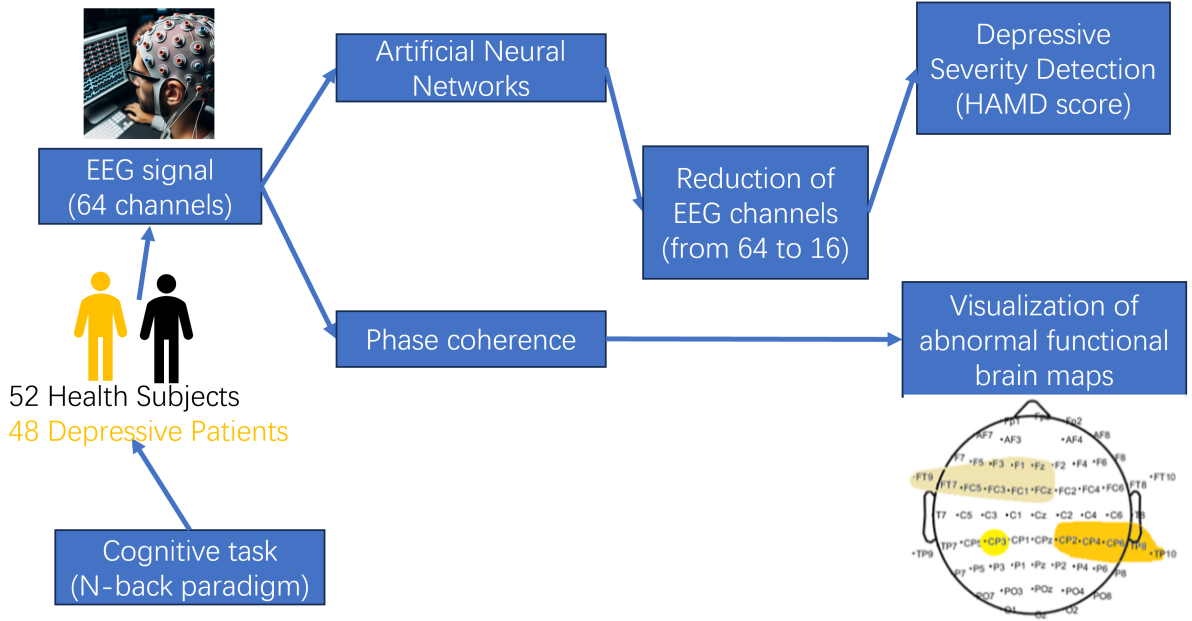


Figure 1.2: The framework of depressive severity scoring system.

Depression is widely categorized as non-depressed, mild, moderate, and severe, according to the severity of the depressive symptoms [30]. However, a descriptive study has shown that the rate of misdiagnosis of [MDD](#) is as high as 65.9% [19]. This means that the primary accuracy rate is less than 35% [19]. Failure to correctly diagnose [MDD](#) is caused by inadequate training of clinicians, as well as reasons that patients are not given appropriate appointments, medical examinations and proper treatments at the early stage [19], [31]. The techniques used for depressive disorder detection can be divided into three rough categories: (1) questionnaires, (2) clinical sensors (such as, [Magnetic Resonance Imaging \(fMRI\)](#), [EEG](#) and [Functional near-infrared spectroscopy \(fNIRS\)](#)) and (3) ubiquitous sensors (such as, accelerometer sensors, WIFI, GPS and so on). There are three most popular questionnaires: the [DSM-IV Axis I Disorders, Clinician Version \(SCID-CV\)](#) [32], the [Hamilton Depression Rating Scale \(HAMD\)](#) [33], and the [Beck Depression Inventory \(BDI\)](#) [34], and all these three have strong histories of use in the psychological sciences. However, this detection method heavily relies on the knowledge and experience

of psychologists. Existing tools used by psychologists and physicians for diagnosing MDD have three main challenges:

(1) they are time-consuming and need to be administrated by well-trained engineers or by professional clinicians [35], [36]; (2) they cannot score depressive severity [37], [38]; (3) there is no interpretable result provided, for example, brain topological maps for the visualization purpose. The challenge is now becoming to provide a new clinical biomarker, to ensure the precision and a quick response. We will discuss this topic in Chapter 2.

Challenges of Non-invasive Visualization Techniques

Non-invasive brain functional network visualization techniques are methods used to map and study the connections and activity within the brain without the need for surgical intervention. These techniques are crucial in understanding brain function, diagnosing neurological disorders, and monitoring treatment effects. Here are some of the key non-invasive techniques, along with their disadvantages.

fMRI: While functional magnetic resonance imaging (fMRI) offers high spatial resolution and detailed brain activity maps, it has several limitations in clinical and research contexts. These include high cost, both in terms of equipment and maintenance, and poor temporal resolution, as fMRI detects blood oxygenation levels that change slowly in response to neuronal activity. This makes it difficult to capture fast neuronal processes, which are crucial for understanding dynamic brain activity, especially in mental health disorders like depression. Additionally, fMRI has contraindications, as it is unsuitable for patients with metal implants, pacemakers, or severe claustrophobia. In the context of depression severity, fMRI has been used to identify biomarkers such as altered connectivity in regions like the prefrontal cortex and amygdala, which are linked to mood regulation [39], [40]. However, these markers are not always specific or consistent across individuals, limiting its use in personalized diagnosis.

EEG: Electroencephalography (EEG) offers several advantages, including low cost, portability, and excellent temporal resolution, making it well-suited for tracking the dynamics of brain activity in real time. However, it suffers from poor spatial resolution, which limits its ability to pinpoint the exact location of brain activity. EEG signals are also highly susceptible to noise and artifacts, such as those from muscle activity, eye movements, or electrical interference. Additionally, EEG primarily captures cortical activity, meaning it may not provide insights into deeper brain structures that play a role in depression, such as the hippocampus or subcortical regions [41]. Despite these limitations, EEG has been widely used in depression research, with studies showing altered brain wave patterns, such as increased theta and decreased alpha activity in the prefrontal cortex, that correlate with the severity of depressive symptoms [42]. Quantifying these patterns could

potentially serve as a means for assessing depression severity and treatment response.

fNIRS: Functional near-infrared spectroscopy (**fNIRS**) provides a less invasive, portable, and relatively low-cost alternative to **fMRI**, offering real-time monitoring of brain activity through the measurement of oxygenated and deoxygenated hemoglobin in the cortical regions. However, it has limited spatial resolution compared to **fMRI**, as it primarily captures signals from the cortical surface and cannot assess deeper brain structures involved in depression, such as the amygdala or hippocampus. Additionally, its depth penetration is restricted, limiting its utility in understanding brain activity in more complex regions of the brain. **fNIRS** is also vulnerable to motion artifacts, which can distort readings, especially in patient populations where movement may be more common (e.g., elderly or pediatric groups). Despite these challenges, **fNIRS** has shown promise in identifying brain activity patterns associated with depression, particularly in the prefrontal cortex. Research has demonstrated that **fNIRS** can detect changes in brain oxygenation levels that correspond with the severity of depressive symptoms, which may offer a potential marker for monitoring treatment efficacy [43], [44].

Each non-invasive brain functional network visualization technique offers unique strengths and weaknesses. **fMRI** provides detailed spatial information but at a high cost and with limited temporal resolution. **fNIRS** offers portability and safety but struggles with depth and resolution. **EEG** excels in temporal resolution but have limitations in spatial accuracy. The choice of technique depends on the specific research or clinical needs, balancing the trade-offs between spatial and temporal resolution, cost, safety, and practicality. **EEG** source analysis [45] provides a noninvasive way to construct the directional brain networks which can visualize the brain dynamic activity under various tasks. In Chapter 3, we will discuss this topic.

1.2.2 Psychotherapy Using Large Language Models

Challenges of Fine-tuning Language Models

Fine-tuning, the process of updating the parameters of pre-trained **Language Models (LMs)**, has demonstrated effectiveness across a variety of downstream **NLP** tasks [46]–[48]. However, traditional fine-tuning methods often encounter inefficiencies due to redundant parameters in fully pre-trained models, which can impede adaptation to new tasks [46], [49]. To address this issue, previous research has explored methods that adapt only specific vectors or introduce additional parameters while keeping most of the pre-trained parameters fixed. This approach enhances operational efficiency by loading task-specific parameters associated with the pre-trained models prior to deployment. **Low-Rank Adaptation (LoRA)** [48] successfully achieves this goal by mitigating inference latency, which in

turn helps extend model depth or reduce the usable sequence length in models [47], [49], [50], striking a balance between efficiency and quality. The challenges in fine-tuning pre-trained LMs for **Natural Language Understanding (NLU)** downstream tasks involve reducing the number of tuned weights and accurately approximating the update of pre-trained weights derived from **LMs** [46]–[49]. Effectively selecting relevant knowledge from pre-trained **LMs** is crucial to overcoming these challenges. This raises the question: why can't we directly suppress "redundant" knowledge during fine-tuning while preserving relevant information?

In the prior work of **LoRA** [48], authors only used the similarity matrix to compare the difference between **LoRA** fine-tuning and fully fine-tuning methods. There is no straight forward visualization result that can show us which part of attention weights has been tuned by such methods. In addition, when using **LoRA** fine-tuning method on **LMs**, we found that although the low rank "bottleneck" can compress information and reweight the pre-trained parameters, such compressed information always contains noise and task-irrelevant knowledge. In Chapter 4, this topic will be discussed and solved.

Challenges of Developing Psychotherapy Chatbots Using Large Language Models

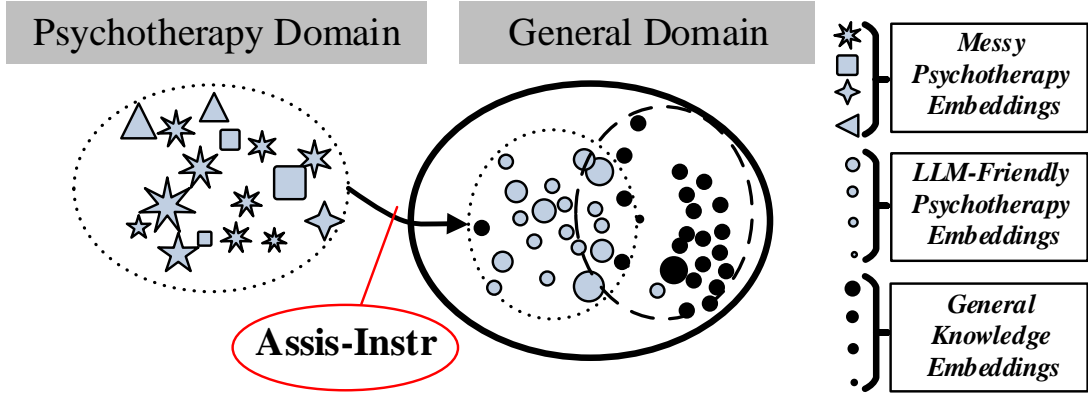


Figure 1.3: A semantic graph that describes how Assistant-Instruction can change the professional embedding to a common embedding. A successful model is expected to use the provided instructions (including task and domain definition examples) to response to professional evaluation.

LLMs have demonstrated impressive generalization capabilities, such as in-context learning [51], chain-of-thoughts reasoning [52], and biomedical diagnosing [53]. Instruction-tuning of **LLMs** has enabled them to follow natural language instructions and perform real-world tasks [54]. Two main methods have been developed for instruction-tuning **LLMs**: (1) fine-tuning the model on a wide range of tasks using human-annotated prompts and feedback [55], and (2) supervised fine-tuning using public benchmarks and datasets

augmented with manually or automatically generated instructions [56]. [Reinforcement Learning on Human Feedback \(RLHF\)](#) has proven to be an effective way to improve [LLMs](#) in various domains, such as medicine [57], knowledge graphs [58], or biomedical applications [59], but it comes with a high cost. Natural instructions [54], and even un-natural instructions [60], can provide knowledge in multiple domains, but [LLMs](#) pre-trained on vast corpora (e.g., Llama1 [61], Llama2 [62] and Llama3 [63] containing books, common crawled conversations, arxiv articles, GitHub, C4, and Wikipedia data) still require additional professional knowledge, especially from domain experts. Self-Instruct tuning [64], [65] and Guess-Instruction tuning methods have shown better performance in aligning [LLMs](#) with human intent by learning from instruction-following data generated by state-of-the-art instruction-tuned teacher [LLMs](#) (e.g., GPT-3, GPT-3.5, and even GPT-4). These lines of instruction-tuning research have proven effective in improving the zero and few-shot generalization abilities of [LLMs](#).

We would like to propose one dataset that contains knowledge of professional psychotherapy and knowledge in the common domain of the pretrained [LLMs](#). This dataset will include a set of instructions, denoted as I_t , where each instruction specifies a particular domain t using natural language. Each domain t contains n_t or more input-output examples $\{(X_{t,i}, Y_{t,i})\}_{i=1}^{n_t}$. Our hypothesis is that each domain t has unique characteristics, as illustrated in the left panel of Figure 1.3. The goal is for a model M to produce the correct output based on the domain-specific instruction and related input, following $M(I_t, X_{t,i}) = Y_{t,i}$ for $i \in \{1, \dots, n_t\}$. In practice, instructions are structured as prompts such as “Provide suggestions or comments on addressing and alleviating the following topic,” with instance inputs such as “addictive disorders.” Sometimes, the boundaries between the instruction and input from instances might blur. For instance, with instructions like “Summarize the description and explain the following concept in the [***] domain, adding relevant background knowledge,” and input instances such as “Addiction and Spiritual Crisis,” the instruction domain can overlap with other domains.

Specific professional knowledge may not always fit cleanly within instructions or outputs, as overlapping domains can introduce unrelated information, potentially destabilizing the training process. To enhance diversity and robustness in the data format, we allow additional knowledge and fine-tuning adjustments from external models (for example, setting $Y = Y + Y'$, where Y' is revised by GPT-4 and incorporated back into the output). The right panel of Figure 1.3 illustrates the challenge of making data LLM-friendly. Thus, in Chapter 5, we use [LLMs](#) to structure instructions, inputs, and outputs, enhancing clarity and consistency.

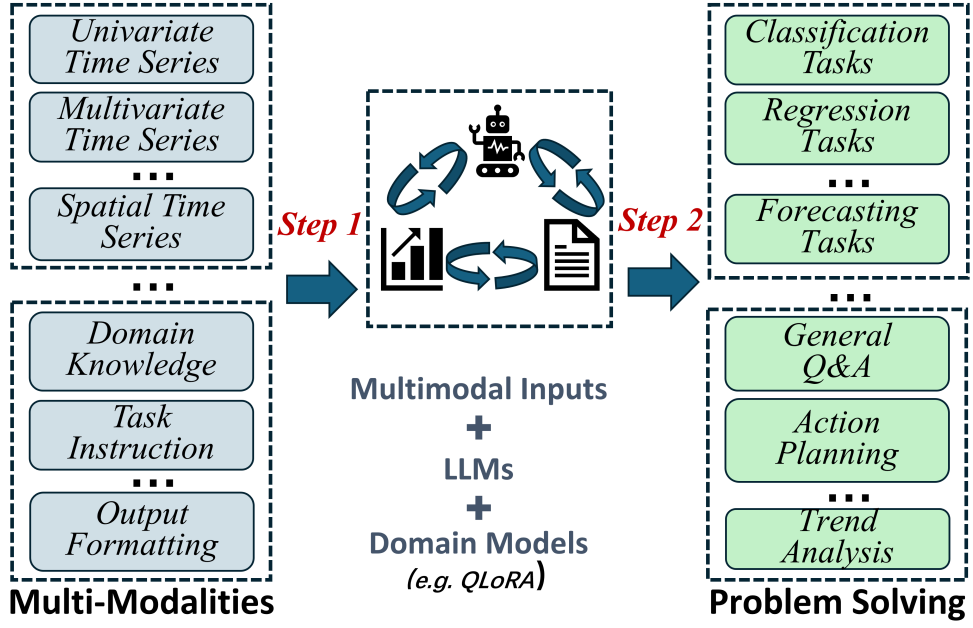


Figure 1.4: The integration of time series and LLMs demonstrates potential in solving complex real-world problems.

Challenges of Large Language Models on Time Series

With the integration of LLMs, time series analysis is undergoing significant transformation [66], [67]. Time series models are conventionally designed for specific tasks, and they depend heavily on prior domain knowledge and extensive model tuning. Existing studies, such as [66]–[68], lack assurances of effective knowledge updates and validations on specific time series tasks.

By aligning time series and natural language, large language and specialistic time series models constitute a new technology paradigm, where the LLMs is prompted with both time series and text-based instructions [69]. In this paradigm, time series and textual information provide essential contexts, LLMs contribute internal knowledge and reasoning capabilities, and time series models offer fundamental guarantees of pattern recognition. This novel integration is depicted in Figure 1.4, where the successful fusion of these components showcases the potential of a general-purpose unified system in next-generation time series analysis. Therefore, the challenge is to develop one tool that can transform the internal patterns of time series to the contents that LLMs can recognize (the Step 1 of Figure 1.4). Moreover, this tool should also transform the generated contents back to time series (especially on time series forecasting tasks), so as to offer the time series analysis assistant (the Step 2 of Figure 1.4). Symbolic approximation methods, such as SAX [70] and ABBA [71], [72] offer an potential way to achieve it. Then, we review existing methods and discuss the potential solution in Chapter 6.

1.3 Thesis Outline

In this thesis, I aim to explore the topics posed above, with the goal of broadening our understanding of both depressive severity detection and the use of **LLMs** in depression psychotherapy—an area that remains relatively unexplored. Chapter 2 examines how we use brain signals to classify depression and assess its severity. Chapter 3 investigates brain activity during tasks involving memory and concentration by using a non-invasive technique, which can be affected by depressive symptoms. Chapter 4 introduces a novel fine-tuning method for enhancing **LLMs** to offer more accurate therapeutic support. Chapter 5 focuses on developing psychotherapy chatbots trained on public therapeutic conversations. Chapter 6 delves into the ability of **LLMs** to analyze time series data. The structure of this thesis is primarily based on the following five publications:

1. **Kang, C.**; Novak, D.; Yao, X.; Xie, J.; Hu, Y#. (2023). "Classifying and Scoring Major Depressive Disorders by Residual Neural Networks on Specific Frequencies and Brain Regions," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 2964-2973, 2023, doi: 10.1109/TNSRE.2023.3293051.
2. **Kang, C.***; Li, Y.*; Novak, D.; Zhang, Y.; Zhou, Q.; Hu, Y#. (2020). "Brain Networks of Maintenance, Inhibition and Disinhibition During Working Memory," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 7, pp. 1518-1527, 2020, doi: 10.1109/TNSRE.2020.2997827.
3. **Kang, C.#**; Prokop, J.; Tong, L.; Zhou, H.; Hu, Y.; Novak, D. (2024). InA: Inhibition Adaption on Pre-trained Language Models. *Neural Networks*, 178, 106410. <https://doi.org/10.1016/j.neunet.2024.106410>
4. **Kang, C.#**; Novak, D.; Urbanova, K.; Cheng, Y.; Hu, Y. (2024). Domain-Specific Improvement on Psychotherapy Chatbots Using Assistant. 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops, Seoul, Republic of Korea, 2024, pp. 351-355, doi: 10.1109/ICASSPW62465.2024.10626529.
5. Carson, E.; Cheng, X.#; **Kang, C.#**. (2024). LLM-ABBA: Large Language Models Understand Time Series Via Symbolic Approximation. (Under review on *IEEE Transactions on Signal Processing*).

In Chapter 2, we constructed abnormal brain network connections related to depression using **EEGs** signals. Based on these abnormal connections, we examined the feasibility of using **EEGs** to detect depression and assess the severity of the depression. To validate this method, we collected **EEG** data from 52 healthy and 48 depressed participants from one

university and one hospital, now publicly available to researchers. We found that [EEG](#) signals from the beta band were particularly effective for depression classification, and the selected channels performed better in scoring depressive severity. This chapter also revealed distinctive brain connectivity patterns, particularly the increased delta deactivation coupled with strong beta activation as depression severity increases. We conclude that the model developed in this chapter is reliable for classifying depression and assessing depressive severity, offering physicians a valuable topological map and a quantitative assessment of depressive severity based on [EEG](#) signals.

In Chapter 3, we used [Phase Lock Coherence \(PLC\)](#) and [General Partial Directed Coherence \(GPDC\)](#) to analyze connections among four adaptively fitted [EEG](#) sources, applying previously published models to describe brain circuits involved in maintenance, inhibition, and disinhibition. Using the classical n-back visual task, we recruited 45 mental health undergraduates and found that the bilateral [Prefrontal Cortex \(PFC\)](#) plays a crucial role in cognitive components such as rehearsal, inhibition, and disinhibition. These findings suggest that the maintenance circuit helps sustain positive cognitive components, inhibition reduces energy consumption by halting repetitive functions, and disinhibition activates new brain activity to focus on novel tasks.

In Chapter 4, we proposed a novel fine-tuning method, [Inhibition Adaption \(InA\)](#), inspired by neuroscience. The [InA](#) method reduces the number of tunable parameters while reweighting knowledge from pre-trained [LMs](#) through an inhibition mechanism. By inserting a small trainable vector into the Transformer attention architecture and setting thresholds to eliminate irrelevant knowledge, we found that [InA](#) outperforms other fine-tuning methods on large models such as $BERT_{large}$, $RoBERTa_{large}$, and $DeBERTa_{large}$ for tasks like text classification and question answering.

In Chapter 5, we proposed a domain-specific instruction tuning method for [LLMs](#) in psychotherapy. By fine-tuning pre-trained [LLMs](#) on a dataset of therapy conversations, we showed that our Assistant-Instruction approach improved the linguistic quality of responses compared to existing [State-Of-The-Art \(SOTA\)](#) models. We also released our large synthetic dataset to facilitate future research on instruction tuning for [LLMs](#) in psychotherapy.

In Chapter 6, we introduced a time-series compression method, [Large Language Model with Adaptive Brownian Bridge-based symbolic Aggregation \(LLM-ABBA\)](#), which enables [LLMs](#) to analyze time series signals. By representing time series patterns through symbolic approximation, we avoid training embeddings from scratch and achieve [SOTA](#) performance on tasks such as time series classification, regression, and forecasting. Extensive experiments on well-established datasets demonstrate the advantages of [LLM-ABBA](#).

This thesis concludes with an overarching framework for conceptualizing the detection

of depressive severity and suggests potential future research directions to advance recovery from depression.

Chapter 2

Classifying and Scoring Depressive Disorders

MDD can be evaluated using both advanced neurocomputing methods and traditional machine learning techniques. This study aims to develop an automated system that utilizes EEG signals to classify depressive states and score depressive severity based on specific frequency bands and electrode data. Two Residual Neural Network (ResNet)-based models—one for classification and the other for regression—are presented, with EEG monitoring applied to classify depression and score severity levels. Key frequency bands and brain regions are selected to optimize ResNet performance. The algorithm, evaluated via 10-fold cross-validation, achieved an accuracy range of 0.371 to 0.571, with Root-Mean-Square Error (RMSE) between 7.25 and 8.41. After focusing on the beta frequency band and 16 specific EEG channels, the system reached a classification accuracy of 0.871 and an RMSE of 2.80. Our results indicate that the beta band provides more distinct features for depression classification, while the selected channels enhance the performance for scoring depressive severity. Additionally, phase coherence analysis revealed distinctive brain network features, including increased delta deactivation and stronger beta activation as depression severity increased. The model developed here provides a practical tool for classifying depression and scoring its severity, offering physicians a method that combines topological brain network analysis with quantified depressive symptoms. The selected frequency bands and brain regions significantly improve depression detection and severity scoring.

2.1 Introduction

MDD is a severe mental illness that significantly increases the risk of suicidal ideation and behaviors [17]. Individuals with depression often face challenges in receiving an

accurate diagnosis, which can lead to issues such as self-medication, substance abuse, inadequate treatment, social isolation, and impaired academic or professional performance [18], [19]. CBT is effective for mild depression, while a combination of psychotherapy and antidepressant medication is the most effective treatment for severe cases [20]–[22]. However, inadequate treatment can lead to relapse or the persistence of discontinuation symptoms [23].

Depression severity is commonly categorized as non-depressed, mild, moderate, or severe [30]. However, studies have shown that the misdiagnosis rate for MDD can be as high as 65.9% [19], implying that the true diagnostic accuracy is less than 35% [19]. This misdiagnosis stems from insufficient clinician training, inadequate early-stage evaluations, and limited access to timely treatment [19], [31]. Current diagnostic tools for MDD face several challenges, such as being time-consuming, requiring specialized training for administration, lacking the ability to classify severity, and not providing useful visualizations like brain topological maps [35], [36].

To address these challenges, this study hypothesizes that delta and beta brain activities are correlated with depression, as indicated by previous research [2], [5], [6], [73], [74]. In pursuit of early depression detection, we analyze delta and beta brain activity along with associated brain networks, visualizing the results. The Phase Synchrony Index (PSI) [2], [5], [6], [73], [74] is computed to construct brain functional networks, selecting relevant electrodes and frequency bands based on differences in PSI between depressed and healthy groups. A ResNet classifier is then used to process the selected EEG signals for depression detection. Additionally, a ResNet regression model is proposed to score depressive severity. The optimized ResNets for EEG signals are designed to accelerate computation and diagnosis, making this system a valuable tool for depression detection, severity monitoring, and evaluating conventional treatments in clinical settings.

Contributions:

1. Demonstrating central-parietal increased delta deactivation and strong beta activation in the severe depression group during working memory tasks.
2. Proposing a ResNet-based classification model with specific frequencies and brain regions for more accurate depression detection.
3. Introducing a ResNet regression model that scores depressive severity based on professional psychologist labels.

The corresponding code and documentation for this study can be found at: <https://github.com/ChengKang520/Classifying-and-Scoring-MDD-BCI>.

bands and electrodes from functional brain networks, detecting depression-related features, and implementing [ResNet](#) models for classification and regression tasks. The classifier outputs a depression diagnosis, while the regression model scores the severity of depression.

2.2.1 Brain Regions and Functional Network Extraction

Methods involving functional or structural brain networks are essential in diagnosing mental health conditions such as bipolar disorder and schizophrenia, with a particular focus on depression detection through [EEG](#) analysis [75]–[77]. Researchers have placed considerable emphasis on the extraction of relevant features during preprocessing to enhance the accuracy of depression detection. In the initial phase of network construction, indices that measure interconnections or spectral characteristics between brain regions are computed. For example, spectral coherence has been used in conjunction with Adaboost classifiers to identify depressive symptoms based on brain regions during the resting state [78]. The left and right frontal-prefrontal regions have shown distinct advantages in identifying depression [79]. Similarly, the absolute power of the theta wave has been a reliable indicator for distinguishing depressive patients from controls, with classifiers like [K-Nearest Neighbor \(KNN\)](#) being employed for this task [75]. Brain networks are known to exhibit abnormal cognitive patterns in depressive patients, such as disruptions in the cognitive control network [80]. Additionally, these networks exhibit characteristic electrophysiological signatures in various frequency bands (e.g., delta, theta, alpha, and beta) [6]. To construct functional brain networks, the [PSI](#) [2], [5], [6], [73], [74] is computed to quantify the synchronization between [EEG](#) channels. This is followed by a correlation-based clustering approach to build convergent brain networks.

In terms of signal processing, Morlet’s wavelet transform is employed to compute the time-frequency domain and phase angle:

$$\varphi_{trialk}^n(f, t) = \frac{1}{\sqrt{\pi}\delta_t} \exp\left(\frac{-t^2}{2\delta_t^2}\right) \exp(j2\pi ft),$$

$$\Delta\theta_{trialk}^{n \rightarrow m} = \text{angle}(\exp(i[\varphi_{trialk}^n(f, t)])) - \text{angle}(\exp(i[\varphi_{trialk}^m(f, t)])),$$

where $\varphi_{trialk}^n(f, t)$ represents the Morlet wavelet at frequency f , and δ_t is the standard deviation of the Gaussian window. These signals are processed using the EEGLAB toolbox within the MATLAB environment, selecting appropriate wavelet cycles and time-frequency windows based on prior research [2], [5], [6].

2.2.2 Utilization of Artificial Neural Networks

Machine learning techniques, particularly [ANNs](#), are pivotal in enhancing the speed and accuracy of depression diagnosis. Machine learning models such as [Support Vector Machine \(SVM\)](#), AdaBoost, and [Random Forest \(RF\)](#) are commonly used in clinical practice with [EEG](#) data. A major challenge in training these models is the potential for overfitting, which can arise if too many irrelevant features are included. To address this, only the most informative channels are selected during the preprocessing stage.

The process of depression detection generally involves three key steps:

1. **Psychological Paradigm:** Cognitive tasks such as the n-back [Working Memory \(WM\)](#) task are frequently used to assess the relationship between cognitive function and depression severity [81], [82]. The n-back paradigm is selected for its ability to manipulate task difficulty and assess [WM](#) capacity, which has been shown to correlate with depressive symptoms.

2. **Feature Extraction:** Relevant features are extracted from neuroimaging regions and electrophysiological signals, often from [EEG](#) channels recorded during resting or task-completion states.

3. **Classification and Scoring:** Machine learning models are trained to classify depressive states and score the severity of depression. Traditional machine learning techniques, such as [ANN](#), logistic regression, [SVM](#), and [Convolution Neural Network \(CNN\)](#), have been used for depression classification. More recently, deep learning methods like [CNNs](#) and [Long Short Term Memorys \(LSTMs\)](#) have demonstrated impressive performance in automating feature extraction and scoring depression severity.

For scoring depression severity, previous studies have utilized [fMRI](#) data and regression models, such as kernel partial least squares regression, to evaluate the severity of symptoms [83]. Our approach, which employs [ResNet](#) models trained on EEG data from the beta frequency band, is aimed at providing an efficient and accurate system for both classification and severity scoring of depression.

2.3 Methodology

2.3.1 EEG Recording and Participants

The [EEG](#) signals used in this study were collected from Shenzhen University and Shenzhen Kangning Hospital, with approval from the ethics committee of Shenzhen Mental Health Center. The dataset consists of 52 healthy undergraduate students (mean age: 20.4 ± 9.7 years) and 48 depressed patients (mean age: 34.3 ± 12.1 years). All participants were screened to ensure they had no history of psychiatric or neurological disorders, and

they were not taking any medications prior to the experiment. Depressive symptoms were assessed using the [SCID-CV](#) and the 17-item [HAMD](#), administered by professional psychologists.

2.3.2 Working Memory Experiments

The n-back task, used to assess working memory, was conducted in the E-Prime 5.0 environment. The 0-back task served as a baseline, while the 1-back and 2-back tasks increased the cognitive load. Participants were presented with letters on a screen and were required to match or mismatch the current letter with previous ones. Reaction times and accuracy were recorded, and [EEG](#) data were collected during task performance. Only correct responses were used in subsequent analyses.

2.3.3 EEG Preprocessing

[EEG](#) signals were preprocessed according to established protocols [2], which included artifact rejection, band-pass filtering (0.16-30 Hz), and baseline correction. Phase coherence analysis was performed prior to training the models. This preprocessing step is essential for the development of an automated system that classifies depression and scores depressive severity using specific frequency bands and electrodes.

2.3.4 Residual Neural Networks

In Figure 2.2, 64 channels recorded [EEG](#) signals over a duration of 2.5 seconds. Subsequently, a down-sampling process reduced the data length from 2500 points to 1250 points. After discarding 98 points from the tail, the input size for the first model was set as $64 \times 64 \times 18$. Two residual neural networks were employed to train the [EEG](#) data for 0-back, 1-back, and 2-back tasks. In the second training phase, 16 electrodes selected using the phase synchronization method resulted in an input size of $16 \times 64 \times 18$. The total size of the [EEG](#) data amounted to 22.5 million sampling points (48 depressive patients + 52 healthy controls) * 60 trials * 3 tasks (0-back, 1-back, and 2-back) * 2.5 seconds * 500 sampling rates = 22.5 million. Testing the [CNN](#) with 6 residual blocks yielded optimal performance, with a parameter size of 0.85 million, effectively preventing overfitting or underfitting issues through proper parameter selection.

Given the widely recognized 65.9% misdiagnosis rate of [MDD](#) [19], we set the detection rate threshold at 70%. Each participant underwent 60 trials, and the depressive probability for a participant was determined by the ratio of trials with predicted probabilities exceeding 70% to the total number of trials. If the predicted probability for a subject on a trial exceeded 70%, the system classified them as 100% depressive for that trial. Finally,

if, during a trial, 33 out of 40 subjects had probabilities from the [ResNet](#) classifier equal to or greater than 70%, the model’s accuracy rate was 82.5% (33/40). Additionally, the second [ResNet](#) regression model provided the severity score of depression, referencing the [SCID-CV](#) system and the [HAMD](#) score.

The Structure of the Residual Nerual Network

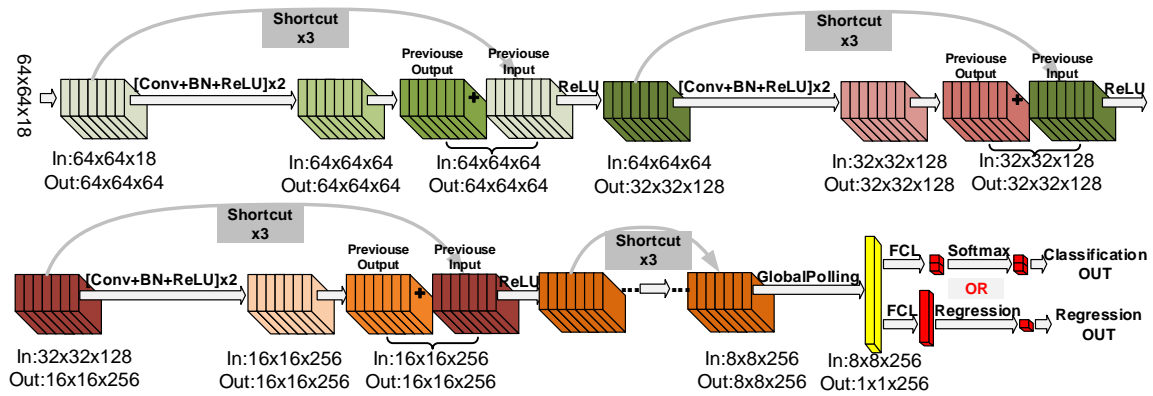


Figure 2.2: The structure of the constructed residual neural network. The input size is $64 \times 64 \times 18$ or $16 \times 64 \times 18$. Conv+BN+ReLU means the processing of convolution (Conv), batch normalization (BN) and rectified linear unit (ReLU). FCL is the fully connected layer. The shortcut is purely forward plus. $\times 3$ means this block should be repeated triple times.

2.4 Result

2.4.1 Memory load comparison of behavioural results

Table 2.1 shows the significant level between the low and the high depressed group in terms of response accuracy rate and reaction time during three different working memory tasks (0-back, 1-back and 2-back). During the 0-back task, there is no significant difference ($P = 0.061$) between MDDs with low scores and the MDDs with high scores in terms of the response accuracy rate. But for the reaction time, the difference is significant ($P = 0.017$). In the 1-back task, both the response accuracy rate and the reaction time show a significant level ($P < 0.01$). When implementing the 2-back task, the MDDs with low scores demonstrated a significant difference in response accuracy rate ($P < 0.01$).

2.4.2 The Connections comparison

We classified the 0-back task as the "rest-state" and the 2-back task as the cognitive load condition for WM. A decrease in PSI reflects reduced neuronal activity in the associated brain regions, suggesting a return to the "rest-state." Conversely, an increase in PSI indicates heightened neuronal activity, reflecting enhanced WM-related processes.

Table 2.1: The comparison of reaction time and response accuracy rates between two different memory loads (average \pm standard deviation) in two depressive groups.

Memory Load		0-back		1-back		2-back	
Scores		Low scores	High scores	Low scores	High scores	Low scores	High scores
Response	Ac-	98.9 \pm 1.4	96.3 \pm 2.2	92.8 \pm 4.7	86.3 \pm 3.6	84.9 \pm 5.3	75.5 \pm 7.6
curacy							
Reaction Time		545 \pm 53	561 \pm 47	701 \pm 147	751 \pm 129	769 \pm 176	791 \pm 183
Statistics		P value		P value		P value	
(the low and the high)		Accuracy	Reaction	Accuracy	Reaction	Accuracy	Reaction
		Rate	Time	Rate	Time	Rate	Time
		P = 0.061	P = 0.017	P < 0.01	P < 0.01	P < 0.01	P = 0.053

Figure 2.3 illustrates the number of significantly connected pairs based on the two WM tasks (0-back and 2-back). Notably, for Ψ decreases, the most pronounced differences across the three groups were observed in the delta frequency components. The high-scoring depressed group predominantly exhibited connections in the theta frequency band, with no significant differences observed in other frequency bands. In contrast, when considering Ψ increases, the high-scoring depressed group displayed the fewest delta, theta, and alpha connected pairs. Both depressed groups exhibited more extensive beta frequency connections, with the low-scoring depressed group demonstrating stronger connectivity in the delta, theta, and alpha bands compared to the high-scoring group.

To assess the overall impact—defined as the product of the number of significant pairs and their corresponding Ψ values—t-values from a two-sample t-test ($P < 0.01$) are presented in Figure 2.4. The most significant frequency component is indicated in each histogram. Except for Figure 2.4B, which shows a moderate increase in delta band activity ($P < 0.05$), the remaining histograms highlight that beta frequency activation contributes most significantly to the observed group differences.

2.4.3 Clusters between these Three Groups

As illustrated in Figure 2.5A, the comparison of Ψ connections reveals a noticeable decrease in Ψ in the depressive group with lower scores, resulting in a sparser electrode connection pattern. This decrease is further reflected in the flat distribution within the beta frequency band shown in Figure 2.3. Conversely, the Ψ increase depicted in Figure 2.5B shows that the control group does not form a cohesive cluster. In contrast, the depressive group with low scores tends to concentrate connected pairs in the left parietal and left central regions, forming what we refer to as Cluster A.

In the comparison between the depressive group with high scores and the control group (Figure 2.5C, lower panel), the Ψ decrease shows that the control group has fewer connected pairs, primarily in the left frontal and whole parietal regions (Cluster C). Meanwhile, the depressive group demonstrates nearly complete cerebral connectivity, with the exception of the occipital areas (Cluster B). Regarding the Ψ increase, the high-

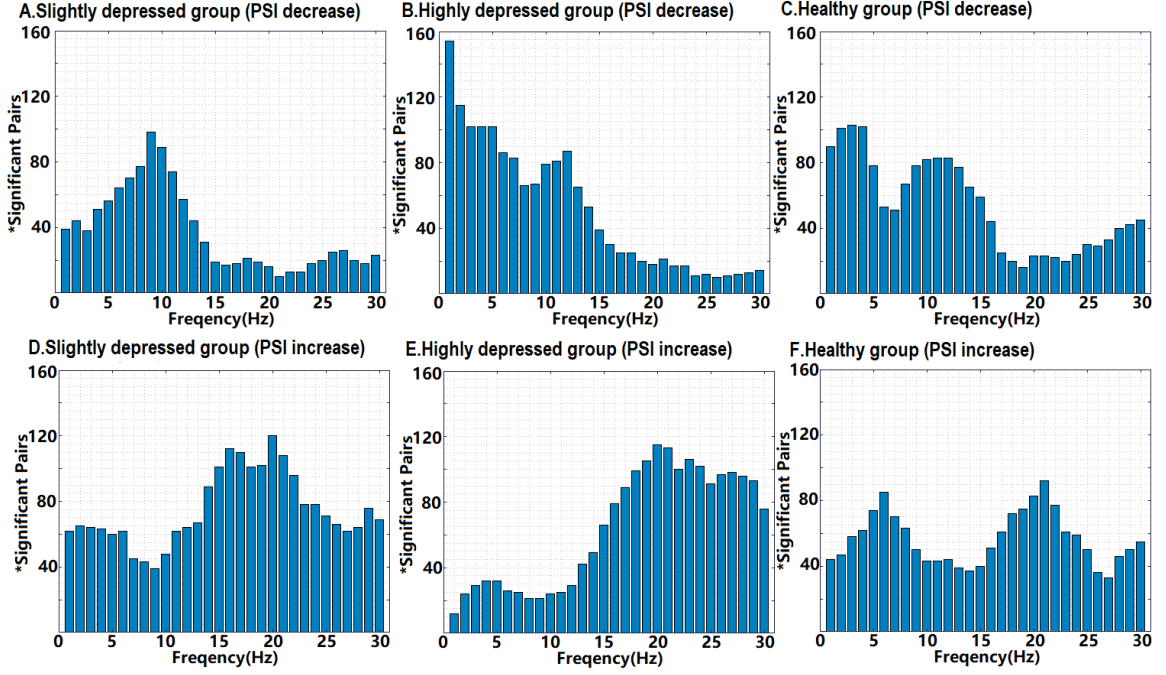


Figure 2.3: The number of the significant pairs in terms of the comparison between 2-back and 0-back tasks.

scoring depressive group exhibits a more compact connection pattern involving the left frontal-central and right central-parietal regions. Additionally, connectivity is observed in the left frontal-temporal and right temporal-parietal regions, forming Cluster D.

Table 2.2: Classification (Accuracy) and scoring depression (RMSE) results (mean and standard deviation) using whole frequency bands (delta, theta, alpha and beta).

Accuracy Rate (>0.70)	0-back	1-back	2-back	Best Result
	0.457 ± 0.063	0.429 ± 0.100	0.514 ± 0.164	0.734 in 2-back
Score Difference (RMSE)	0-back	1-back	2-back	Best Result
	8.38 ± 3.22	8.41 ± 3.52	7.73 ± 3.22	3.22 in 2-back

Table 2.3: Classification (Accuracy) and scoring depression (RMSE) results (mean and standard deviation) using beta frequency bands.

Accuracy Rate (>0.70)	0-back	1-back	2-back	Best Result
	0.514 ± 0.217	0.429 ± 0.226	0.371 ± 0.217	0.783 in 0-back
Score Difference (RMSE)	0-back	1-back	2-back	Best Result
	7.97 ± 2.25	7.59 ± 1.51	8.05 ± 1.40	4.10 in 0-back

2.4.4 Results of Classifying and Scoring MDD Patients

After completing the preprocessing steps, each subject underwent up to 60 trials, with substandard trials excluded. The results across the entire frequency band are summarized in Table 2.2, while the specific outcomes for the beta frequency band are shown in Table

Table 2.4: Classification (Accuracy) and scoring depression (RMSE) results (mean and standard deviation) using whole frequency bands (delta, theta, alpha and beta) and selected EEG channels.

Accuracy Rate (>0.70)	0-back	1-back	2-back	Best
	0.514±0.217	0.514±0.239	0.571±0.141	0.714 in 1-backs
Score Difference (RMSE)	0-back	1-back	2-back	Best
	7.77±3.11	7.25±2.19	7.37±2.14	2.88 in 1-back

Table 2.5: By scaling the size of proposed ResNets, the below shows the classification (Accuracy) and scoring (RMSE) results using beta frequency band and selected EEG channels.

ResNet (Size: 2.4M)				
Accuracy Rate (>0.70)	0-back	1-back	2-back	Best
	0.452±0.302	0.409±0.222	0.414±0.367	0.833 (0-back)
Score Difference (RMSE)	0-back	1-back	2-back	Best
	8.12±3.38	8.07±3.44	7.74±3.66	3.02 (0-back)
Default ResNet (Size: 4.6M)				
Accuracy Rate (>0.70)	0-back	1-back	2-back	Best
	0.429±0.226	0.514±0.126	0.457±0.234	0.871 (2-back)
Score Difference (RMSE)	0-back	1-back	2-back	Best
	7.97±3.57	7.83±3.31	7.59±3.83	2.80 (2-back)

2.3. In the second model, the system was expanded to classify depression and assess depressive severity by focusing on the beta frequency band and selecting 16 significant electrodes. During the online clustering step using [PSI](#), two clusters, Cluster A and Cluster D, emerged. The electrodes most frequently connected in both clusters—Fz, F1, F3, FCz, FC1, FC3, FC5, FT7, FT9, T7, CP3, CP2, CP4, CP6, TP8, and TP10—played a crucial role in enhancing the performance for classifying depression patients and scoring their depressive severity. To minimize variability in the results, a 10-fold cross-validation approach was employed to identify the optimal outcome. For example, Table 2.4 shows a classification accuracy of 0.714 using the entire frequency bands, while Table 2.5 indicates that focusing on the beta frequency band achieves an accuracy of 0.871. Ultimately, for the 2-back task within the beta frequency band and using specifically selected electrodes, the highest accuracy achieved through 10-fold testing was 0.871. Regarding depressive severity assessment, although a minimum [RMSE](#) of 2.8 was achieved in the 2-back task with the beta frequency band and selected channels (Table 2.5), the overall performance for scoring depressive severity (Table 2.5) was inferior to that seen with the entire frequency bands in Table 2.4.

2.5 Discussion

In this study, we define deactivation as the dominance of the rest state, and activation as the engagement of [WM](#) processes. Our results reveal that the low depressive group exhibits weaker delta deactivations but stronger beta activations. In contrast, the high

depressive group shows more pronounced delta deactivations and increased beta activations. As depressive severity increases, there is a notable emergence of beta-related right central-parietal functional connections in patients with depression. Furthermore, beta frequency bands play a significant role in distinguishing depressive patients from healthy controls. Selectively chosen electrodes provide a reliable means of differentiating depressive patients, and the use of beta frequency bands improves the accuracy of scoring depressive severity, with selected channels showing substantial scoring advantages.

2.5.1 Potential Inducing Factors for Depression

As depressive symptoms become more severe, individuals with depression exhibit more pronounced delta deactivations and beta activations. Interestingly, no clear evidence has been found linking theta and alpha oscillations to depressive states. It is noteworthy that individuals infected with [Human Herpesvirus 6 \(HHV-6\)](#) show no significant correlation with theta and alpha [EEG](#) oscillations, as reported by [84]. Furthermore, after a 14-day recovery period following medical intervention, [HHV-6](#)-infected patients demonstrate a noticeable deceleration in theta and delta oscillations [85], suggesting a weakening of these activities [86]. Additionally, [HHV-6](#) infection has been linked to an increased risk of mental disorders, particularly depression [87], [88]. Given these findings, we hypothesize a potential connection between [HHV-6](#) and depression, which will be explored in future research to assess the extent to which [HHV-6](#) may contribute to the development of depressive symptoms.

2.5.2 Topological Analysis

The topological network approach facilitates the comparison of cognitive patterns across subjects. Phase coherence analysis reveals that individuals in the depressive group generally exhibit reduced low-frequency [WM](#) activation, particularly in the delta and theta frequency bands. This trend becomes more pronounced as depressive symptoms progress from moderate to severe. As illustrated in Figure 2.4 C and D, highly depressed patients exhibit a significant disparity in beta [WM](#) activation compared to those in the mildly depressed group. This suggests that the depressive group demonstrates stronger beta activations than healthy controls, with highly depressive patients being particularly susceptible to this imbalance. Mildly depressed patients exhibit deficiencies in delta and theta [WM](#) deactivation, while the highly depressive group displays redundant delta and theta [WM](#) deactivation. During [WM](#) tasks, depressive patients show reduced frontal-midline theta power and increased occipital upper alpha power during [WM](#) encoding [89], aligning with previous research suggesting abnormal brain activity across all frequency

bands in depression.

The topological structure of beta frequencies (Cluster D in Figure 2.5) among highly depressive patients reveals additional central-parietal WM activation compared to the mildly depressed group (Cluster A in Figure 2.5). This finding is consistent with studies indicating that MDD is characterized by unique EEG oscillations in the beta frequency range, which dominate over delta, theta, and alpha frequencies when compared to healthy controls [90], [91]. High beta coherence has been associated with connectivity within and between regions such as the Dorsolateral Prefrontal Cortex (DLPFC) and temporal regions [79].

Increased delta deactivation during WM tasks may reflect low WM load and could be associated with a resting recovery mechanism following cognitive effort. The comparison of Cluster B and Cluster C (Figure 2.5, panel C) along with the increase in delta activity (Figure 2.3) suggests that delta deactivation increases as depressive symptoms intensify, in line with findings from neuromodulation therapy studies [92]. Mildly depressive patients exhibit a lack of delta and theta WM deactivation, whereas highly depressive patients show excessive delta and theta WM deactivation. Additionally, studies have found that increases in beta and gamma power in the Left-Dorsolateral Prefrontal Cortex (L-DLPFC) correlate with improvements in depressive symptoms [92]. Enhanced attentional processes associated with beta and gamma oscillations [93] may help explain how beta oscillations modulate attentional processing in depressive subjects. This is further supported by the comparison between Figure 2.3D (decreased alpha activation) and Figure 2.3E, suggesting that greater reductions in upper alpha and gamma power during WM maintenance are indicative of higher depressive severity [89].

2.5.3 Contribution of Frequency and Topological Selection for Classifying and Scoring Depressive Patients

The use of a ResNet classifier to differentiate depressive patients from healthy controls demonstrated that focusing solely on the beta frequency band yields higher classification accuracy than using the entire frequency range. In assessing depressive severity, the system introduced an effective method for quantifying the severity of depression. This suggests that the beta frequency band holds promise for identifying depression during WM tasks [6]. However, although beta frequency activity can serve as a diagnostic tool for depression, it does not significantly enhance the accuracy of scoring depressive severity.

It is important to note that scoring results within the beta band showed a wide variance. To improve the robustness of depressive severity assessments, it is recommended to consider the inclusion of all frequency bands. The relatively lower average accuracy may

be attributed to the small number of psychologists involved in diagnosing patients—only two in this case—which introduces instability in the data, particularly when testing deep learning models with potentially misdiagnosed subjects.

2.5.4 State of the Art for Classifying Depressive Patients

Table 2.6 demonstrates the advantages of our proposed method, achieving the highest accuracy of 87.1% for detecting depression. However, when assessing the overall performance in scoring depressive severity (Table 2.5), the results are weaker compared to those in Table 2.4, which utilizes the entire frequency bands. This discrepancy may be due to the influence of data quality and the limited robustness of the proposed model. A notable limitation of our method is its inability to consistently yield stable results. Furthermore, the approach relies on psychological paradigms, specifically the n-back task, which primarily captures brain function associated with working memory.

Table 2.6: Comparison with existing methods on classifying depression with EEGs.

References	Subjects	Cross validation	Method + Feature	Accuracy
EEGs (Scenario)				
Hanshu Cai et al (2020)[76]	MDD = 86, HC = 92	10-fold	KNN + EEGs (Fp1, Fpz, Fp2)	Highest at 86.98%
Xiaowei Zhang et al (2020)[94]	MDD = 81, HC = 89	10-fold	CNN + EEGs + demographic	Average at 75.29%
Xiaowei Li et al (2019)[95]	MDD = 24, HC = 24	24-fold	CNN + EEGs (all frequencies)	80.74% for mild
The proposed method	MDD = 48, HC = 52	10-fold	ResNet + EEGs (beta bands 16 electrodes)	Max: 87.1% and Average at 45.7%

2.5.5 State of the art for scoring depressive severities

Scoring of depressive severity is addressed in two studies based on [Magnetic Resonance Imaging \(MRI\)](#)-related images with [Partial Least Squares Regression \(PLSR\)](#) and [Relevance Vector Regression \(RVR\)](#) [96]. Table 2.7 shows that under the leave-one-out cross-validation, the minimum [RMSE](#) can reach 2.50 [97], which means the [RVR+MRI](#) method can precisely grade the depressive severity within 2.50 error. In this study, the proposed method shows a minimum [RMSE](#) of 2.80 under 10-fold cross-validation.

2.6 Conclusion and future work

In this study, we developed a system consisting of two models based on the [ResNet](#) architecture. The first model is designed for depression detection, while the second model assesses the severity of depressive symptoms. Both models utilize 16 carefully selected

Table 2.7: Comparison with existing methods on scoring depressive severities with EEGs.

References	Subjects	Cross validation	Method + Feature	RMSE
Images (Scenario)				
Kosuke Yoshida et al (2017)[83]	MDD = 58, HC = 65	leave-one-out	PLS + sMRI	9.56
Benson et al (2012)[97]	MDD = 30, HC = 0	leave-one-out	RVR + MRI	2.50
EEGs (Scenario)				
The proposed method	MDD = 48, HC = 52	10-fold	ResNet + EEGs (beta bands 16 electrodes)	2.80

EEG channels and focus on beta frequency signals. The ResNet classifier is used to distinguish depressive subjects from healthy controls, while the ResNet regression model quantifies the severity of depression. Coherence analysis was employed to identify key frequency bands and functional brain networks associated with depression, with a particular emphasis on the role of beta frequency in both detecting depression and scoring its severity. The selected EEG channels demonstrated significant advantages in classifying depression. Although the model developed in this chapter shows promising results for depression classification and severity assessment, it is crucial to validate its performance using external datasets. This would help assess its generalizability and highlight the importance of testing the proposed model in diverse contexts. Future research will focus on: 1. Advancing the construction and optimization of ANNs. 2. Refining EEG data acquisition methods and selecting more representative participants with depression. 3. Designing more robust experiments to validate the model's efficacy. 4. Investigating the impact of antidepressant treatments on EEG signals. Additionally, we plan to explore potential connections between the inducing factors of depression and HHV-6, which could provide insights into underlying mechanisms of the disorder.

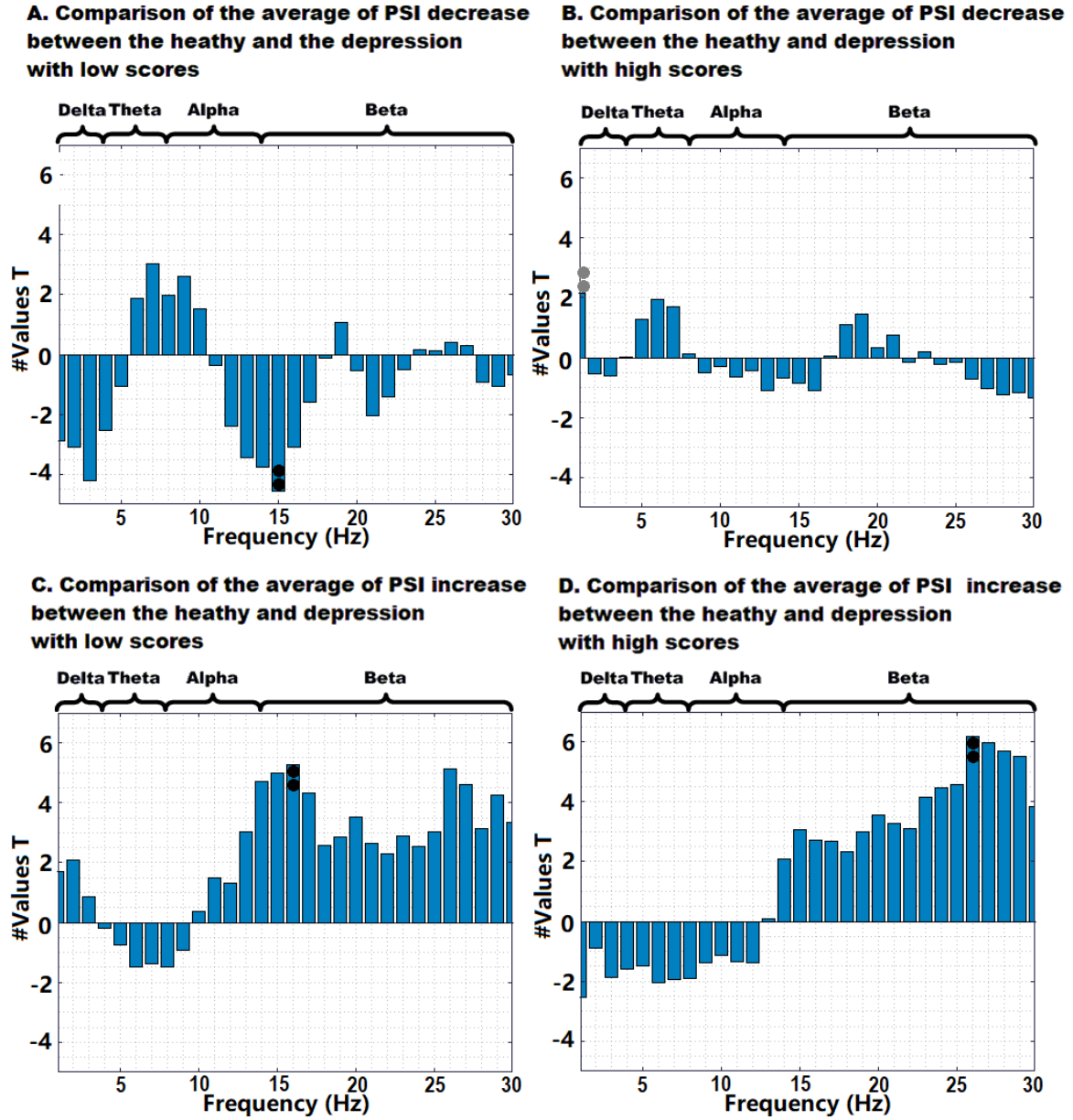


Figure 2.4: The t values (significant level) of the comparison between the depression group and the healthy control group.

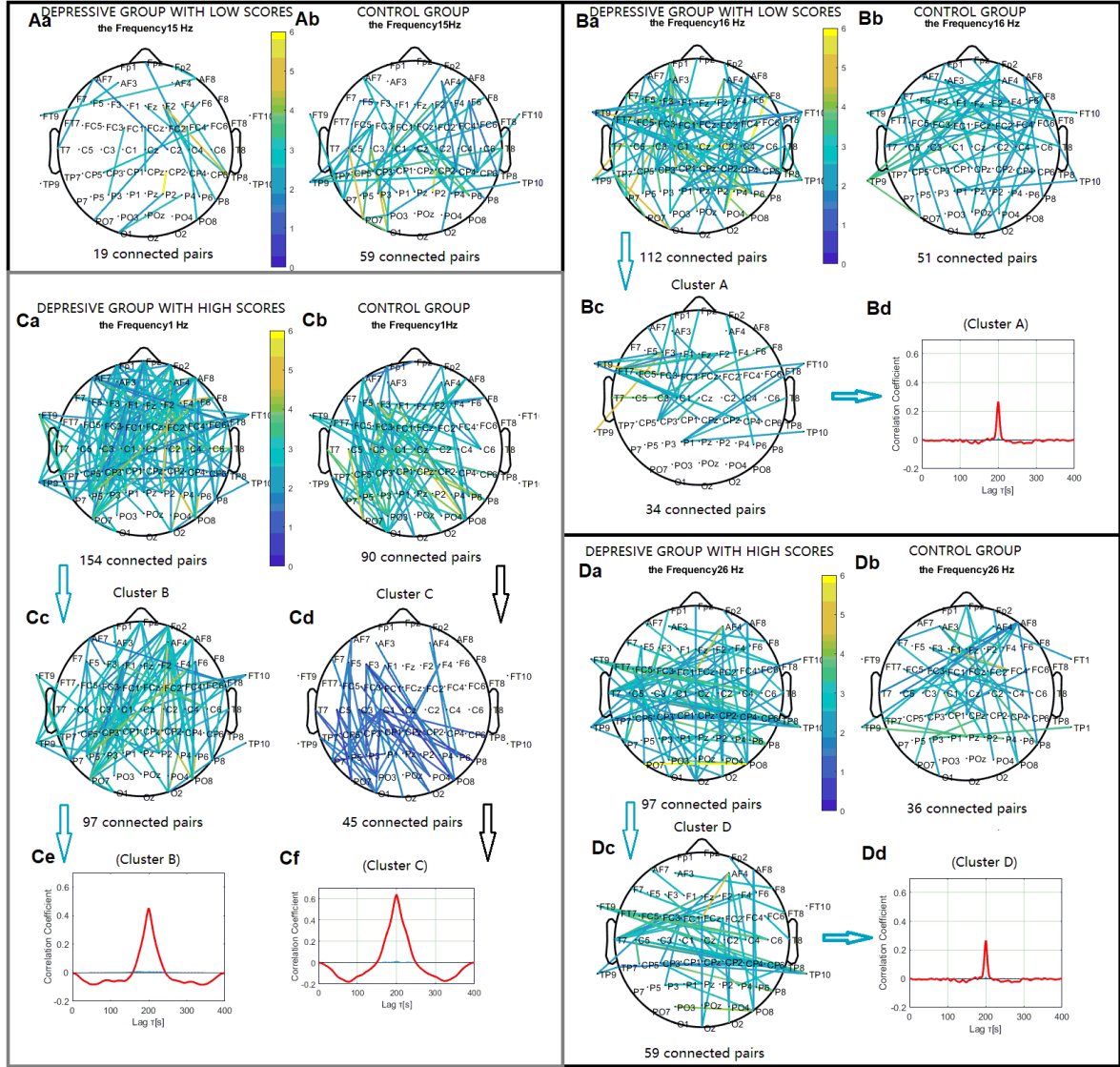


Figure 2.5: Clustering of significantly increased and decreased phase synchronization indices primarily in the beta bands for both depressive groups and the control group. The upper panels (A and B) show the significant PSI decrease and increase during the 2-back task, compared to the 0-back condition ($p < 0.05$) between the depressive group with low scores and the control group. The lower panels (C and D) show the significant PSI decrease and increase between the depressive group with high scores and the control group. Clusters A, B, C, and D represent significant groupings identified with a family-wise error rate correction at $\alpha = 0.01$. The panels labeled Bc, Cc, Cd, and Dc show correlation coefficients for phase synchronization within the corresponding clusters. The gray panel (C) indicates that the significance level is slightly weaker.

Chapter 3

Noninvasive Visualization of Brain Networks

WM is a critical cognitive function responsible for the temporary maintenance and manipulation of information, serving as a key indicator of brain function. The processes involved in memory retention, inhibition, and disinhibition are central to understanding the brain's neurocognitive architecture. Although several theoretical models have been proposed to elucidate the complex WM process, comprehensive evidence detailing the specific brain regions and structures involved in maintenance, inhibition, and disinhibition remains limited. In our study, we applied phase-lock coherence and general partial directed coherence to investigate interactions among four adaptively fitted EEG sources. Additionally, we leveraged previously established models to map brain circuits associated with memory maintenance, inhibition, and disinhibition. The experiment, conducted with forty-five mental health undergraduates using a classical visual n-back paradigm, revealed key insights into the role of the brain in WM. Notably, the bilateral PFC was found to be primarily involved in cognitive functions such as rehearsal before recognition for object classification, inhibition to maintain positive memory, and disinhibition to stimulate subsequent brain interactions. Our findings also indicated that the right PFC occasionally assisted the left PFC in managing high-capacity WM tasks. In contrast, posterior regions, specifically the Posterior Parietal Cortex (PPC), were engaged in attention arousal and memory maintenance. These findings led us to the following conclusions: 1. The recurrent maintenance circuit is crucial for executing positive cognitive components associated with WM 2. Inhibition functions temporarily pause sustained cognitive activities, effectively conserving energy. 3. Disinhibition facilitates the next phase of cognitive processing, enabling the selection of new objects or focusing on novel stimuli.

3.1 Introduction

WM is defined as the cognitive ability to maintain and manipulate information over short periods of time [98]. It is closely linked to attentional control [99] and academic performance [100]. While there is no universal agreement on the neurocognitive architecture of **WM**, its core function involves the short-term maintenance and manipulation of information [101]. A variety of traditional **WM** paradigms, typically characterized by lower capacity, have been employed to assess the cognitive performance of individuals with mental impairments, including those diagnosed with schizophrenia, stroke, traumatic brain injury, and **Attention Deficit-Hyperactivity Disorder (ADHD)**. However, there remains a critical, unmet need for non-invasive methods to assess **WM** activity and guide psychological interventions.

This study takes a multifaceted approach to examining **WM**. First, (i) we assess behavioral performance using n-back paradigms. Second, (ii) we analyze brain networks associated with **WM** using phase-lock coherence and directional coherence following the adaptation of a 64-channel **EEG**, with four sources generated to simulate cerebral internal communication. Lastly, (iii) we propose a "neurocognitive architecture" of working memory based on region-to-region connectivity, identifying pathways for memory maintenance and lateral inhibition during **WM** tasks. This study provides insights into the processes of **WM** and the corresponding brain regions through coherence analysis, offering a non-invasive assessment of functional networks during **WM** tasks in the healthy population.

The proposed neurocognitive architecture of **WM** [101] includes the following five key components: 1. Selective attention 2. Object information recognition and maintenance 3. Rehearsal process 4. Updating and attention sustenance 5. Inhibition [101], [102].

This framework integrates various processing descriptions and emphasizes the concepts of memory maintenance and lateral inhibition [101]. Regions such as the visual cortex, **PFC**—specifically the posterior superior frontal gyrus and middle frontal gyrus—the **PPC**, and the inferior temporal cortex, are integral to visual **WM** tasks [98], [103], [104]. This chapter outlines the processes involved in **WM** and the corresponding brain regions through coherence analysis, offering a non-invasive assessment of brain networks during **WM** tasks in healthy individuals. Based on different neurocognitive stages, three major processes during **WM** tasks are explored in this study:

1. Behavioral performance assessment through n-back paradigms.
2. Brain network analysis using phase-lock coherence and directional coherence, based on adaptively fitted 64-channel **EEG**.
3. The proposal of a "neurocognitive architecture" of **WM** based on region-to-region connections, identifying pathways for memory maintenance and lateral inhibition.

3.2 Related Work

3.2.1 Pathway for Attention Arousal and Executive Function

The **PFC** is widely recognized for its pivotal role in the maintenance of information during **WM** tasks. Meta-analyses consistently demonstrate that the left **PFC**, particularly the ventral aspect, is associated with verbal **WM** tasks, while the right **PFC**, especially the dorsal aspect, is activated during spatial **WM** tasks [105]–[108]. Lesion studies further corroborate these findings, showing that electrophysiological activity in the **PFC** of monkeys reflects these functional distinctions [109], [110]. **FNIRS** has also been used to assess **WM** load by monitoring hemodynamic activity in the **PFC** [111], reinforcing the importance of the **PFC** in **WM**. In addition to the **PFC**, the **PPC** is strongly implicated in **WM** processes [112], with spatial **WM** tasks engaging bilateral parietal regions [105], [107]. **fMRI** and **Positron Emission Tomography (PET)** studies have demonstrated that the **PFC** plays a key role in selecting content from posterior regions [108]. Some studies have suggested that the superior parietal cortex is also involved in executive function and selective attention control [113], [114]. Moreover, research into the integrity of white matter pathways has revealed connections between the **PFC**, parietal cortex, and temporal cortex during **WM** tasks [104], [115].

3.2.2 Pathway for Coding and Decoding

Effective **WM** requires the encoding and subsequent selection of relevant information amidst distractors [116]. The dynamic interaction between the **PFC** and **PPC** has been shown to generate top-down signals that modulate stimulus-coding networks [117], [118]. The **PFC**'s adaptive coding is crucial for task-specific learning, as demonstrated by its ability to classify learning tasks [119], [120]. Notably, population coding within **PFC** neurons has been implicated in transitions between different representational states, particularly during delayed paired associates tasks [121].

Source analyses have revealed that initial visual encoding occurs in posterior brain regions, while selection rules are encoded in the **PFC**. These encoding and decoding mechanisms are essential for maintaining memory content [122]. Multivariate decoding and source analyses further show that prefrontal and parieto-occipital persistent oscillatory neural activity are vital for selecting and maintaining memory content [122].

3.2.3 Pathway for Sustained Brain Activity

Maintenance and sustenance in the brain may involve processes such as memory storage, task and goal maintenance, and attention sustenance. Stronger synaptic connectivity is

thought to be associated with brain networks involved in sustained higher activity [101]. Specifically, fronto-parietal activity has been linked to task-general processing, such as maintaining goals and task sets [123].

3.2.4 Pathway for Lateral Inhibition

In WM tasks, inhibition becomes critical when the system approaches its capacity. Inhibition prevents the decay of persistent activity [98]. A dynamic "winner-take-all" model has been proposed to explain lateral inhibition among memory representations, ensuring that only the most relevant memory remains active [101]. Cognitive inhibition (the ability to suppress irrelevant information) and response inhibition (the ability to suppress automatic responses) are crucial for integrating new and old information in WM tasks [124]. While several regions, including the superior parietal cortex and frontal areas, are involved in inhibition during WM tasks [117], [118], [125], few studies have explored the architecture of inhibition across these regions. To validate the "human neurocognitive architecture" of WM, we use EEG sources and their connections to construct a communication model based on these cognitive components. Dynamic and statistical algorithms, such as neural information flow directionality [126]–[128] and PLC [6], [129], have been applied to measure the transmission of neural signals. Techniques like Partial Directed Coherence (PDC) [130] and GPDC [131] have proven effective in analyzing brain networks based on EEG studies, with applications in various clinical populations, including patients with Parkinson's disease [128], Alzheimer's disease [132], depression [5], [6], and hippocampal-prefrontal activation in monkeys [133].

3.3 Methods

3.3.1 Participants

Forty-five healthy undergraduate students (6:4 male to female, mean age = 20.4 years) participated in the visual n-back paradigm tasks. The study was approved by the institutional ethics review board, and informed consent was obtained from each participant. Participants had no history of psychiatric or neurological disorders and were not on any medication prior to the study.

3.3.2 Experimental Procedures

The n-back task was implemented using E-Prime 5.0. The letter-based variant of the n-back task was employed, with the 0-back serving as a baseline and the 2-back as the

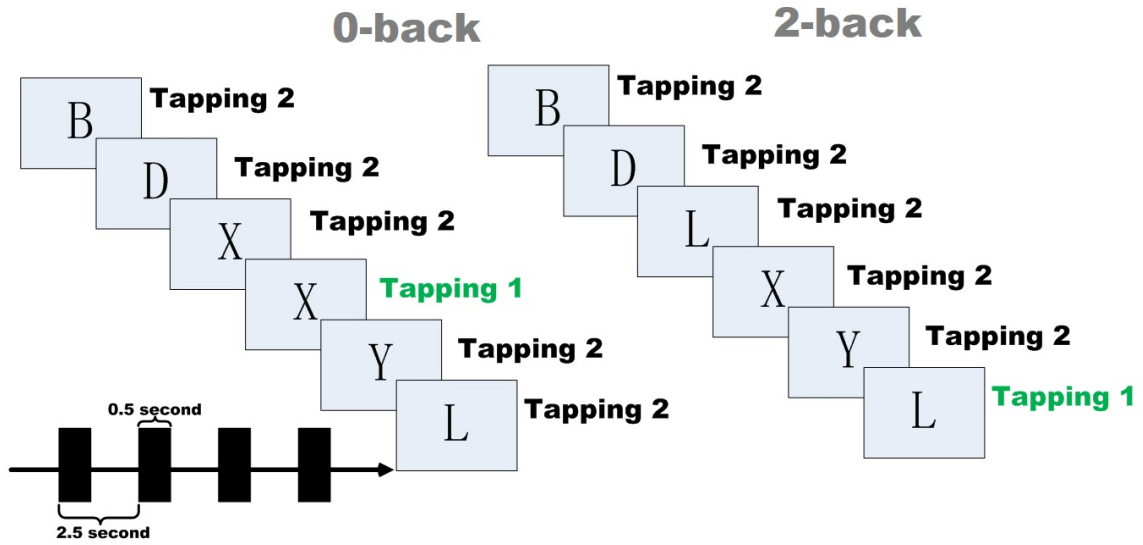


Figure 3.1: Experimental procedure and timeline. Participants responded to stimuli by pressing the '1' key with the index finger for target stimuli (match) and the '2' key with the middle finger for nontarget stimuli (mismatch).

working memory load. In the 0-back task, participants were asked to identify a pre-specified target letter ('X'), while in the 2-back task, they were asked to identify a letter matching the one presented two trials earlier. Stimuli (letters) were randomly selected from the English consonants, as shown in Figure 3.1.

The experiment consisted of three blocks, each containing two 0-back and two 2-back tasks, with the task order randomized. Each task lasted 75 seconds and included a pseudorandom sequence of 30 consonants (10 targets and 20 nontargets). Letters were presented for 0.5 seconds, followed by a 2-second inter-stimulus interval. A 45-second break was given between each block. Participants were instructed to respond as quickly and accurately as possible. Reaction time and accuracy were recorded, with incorrect responses excluded from EEG analysis. Prior to the experiment, participants completed practice trials to familiarize themselves with the task.

3.3.3 EEG Recording

EEG data were recorded using a BrainAmp amplifier (Brain Products, Munich, Germany) and Braincap electrode cap (EASYCAP, Herrsching, Germany) according to the international 10–20 system. The EEG was referenced to the FCz electrode, with the AFz electrode serving as the ground. Vertical and horizontal Electrooculogram (EOG) were recorded from two additional channels placed at the right and left of the eyes. Electrode impedance was kept below 5kΩ. EEG signals were sampled at 1000 Hz with no filtering applied during recording.

3.3.4 Data Analysis

EEG data were preprocessed using a band-pass filter (0.16–30 Hz, 24 dB/octave), followed by artifact rejection and baseline correction. The EEG analysis was divided into preprocessing, source modeling, PLC, and GPDC. For preprocessing, incorrect trials were removed, leaving an average of 53 trials for 0-back and 49 trials for 2-back tasks per subject. These trials were used for subsequent source modeling and coherence analysis. PLC analysis was used to assess the stability of phase synchrony between sources, while GPDC was used to examine directed connectivity between brain regions.

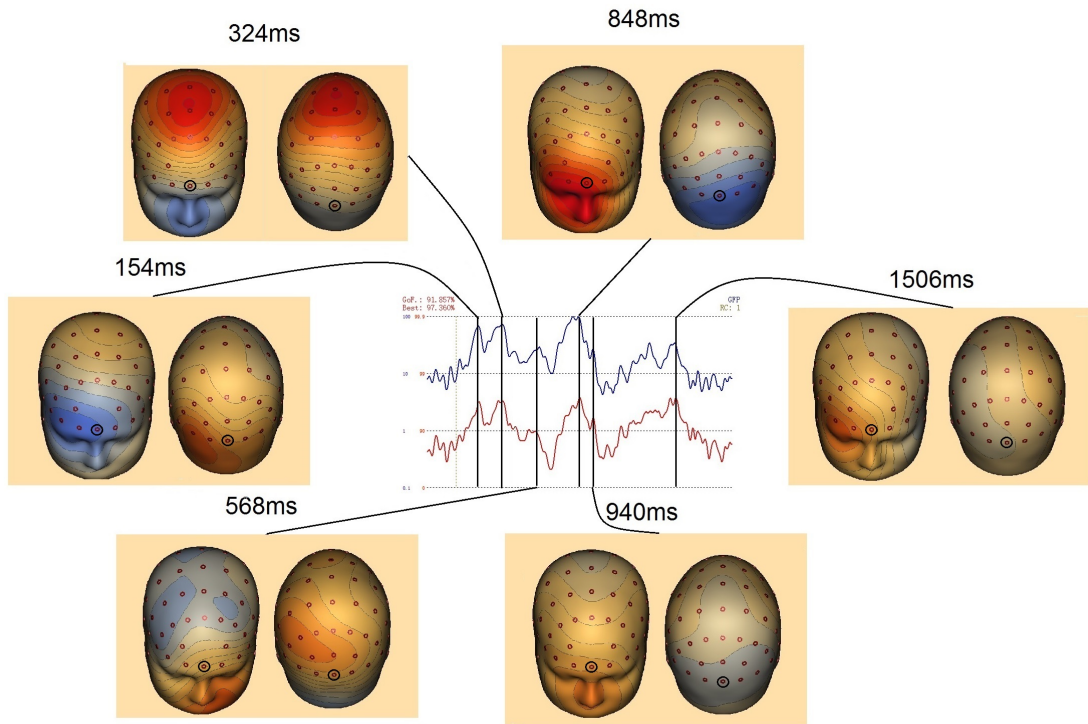


Figure 3.2: Scalp voltage maps for the 2-back condition minus the 0-back condition, showing distinct activation patterns in the front and back hemispheres during different time periods. The circled electrode sites correspond to Fz and Oz. The Global Field Power, which represents the sum of squared amplitudes across all channels, is shown in a logarithmic scale.

Data Preprocessing and Single-Trial Source Waveform Extraction

For each subject, Evoked Related Potential (ERP) waveforms were averaged for the 0-back and 2-back conditions. The difference between these conditions was computed for each subject, and the collective representation of EEG was generated by averaging across subjects. The scalp topography of these difference waves is shown in Figure 3.2.

Source localization was performed using Brain Electrical Source Analysis (BESA 6.0) software. A realistic head model was used to estimate the source configuration, based

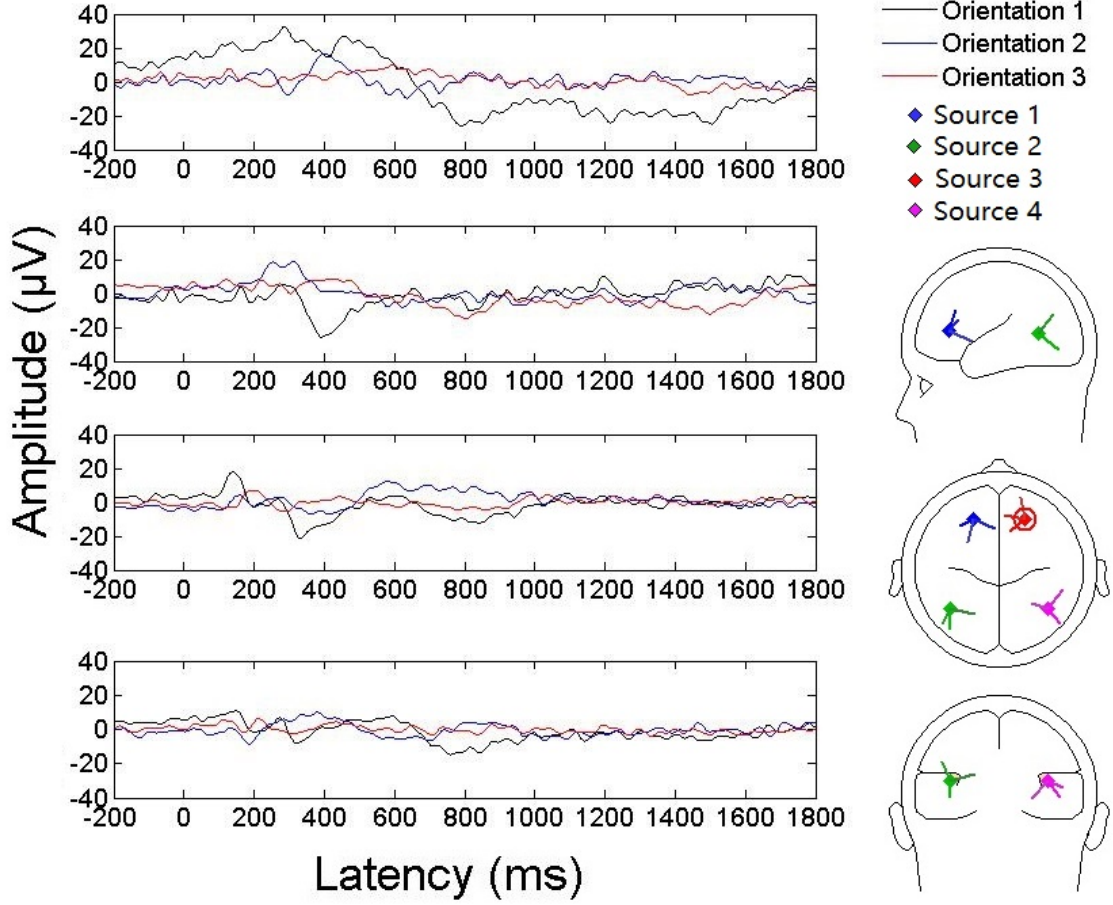


Figure 3.3: RSs and their corresponding time courses of group average EEGs. The left panel shows the three directional time courses of the RSs, and the right panel shows the locations and orientations of the four RSs, with orientation 1 representing the primary orientation of each RS.

on findings from [fMRI](#) studies indicating activations in the bilateral superior/inferior parietal lobules and bilateral inferior frontal gyri during the 2-back vs. 0-back contrast [134]. A [Regional Source \(RS\)](#) model, composed of four sources, was used to estimate the underlying brain activity. The primary orientation of each [RS](#) was set to match the dipole moment of the averaged [ERP](#) difference waves (Figure 3.3). The resulting [RS](#) model was then applied to the [EEG](#) data to extract single-trial source waveforms, which were used for subsequent coherence analysis.

PLC Analysis

PLC was calculated using Morlet's wavelet transform in the time-frequency domain:

$$\omega_{trial,i}(f, t) = \left(\frac{1}{\sqrt{\pi\delta_t}} \exp(-t^2/2\delta_t^2) \exp(j2\pi ft) \right), \quad (3.1)$$

and [Phase Lock Value \(PLV\)](#) was used to quantify the phase synchrony between

different RSs:

$$PLV_{l,m}(f, t) = \left| \frac{1}{n} \sum_{\text{trial}=1}^n \exp(i [\omega_{\text{trial},l}(f, t) - \omega_{\text{trial},m}(f, t)]) \right|, \quad (3.2)$$

where n is the number of trials. The PLV was calculated for each frequency from 1 Hz to 30 Hz. A two-sample t-test was used to assess the differences in PLV between 0-back and 2-back tasks across time and frequency domains. Additionally, 1000 bootstrap resampling was performed to assess statistical significance.

GPDC Analysis

As consistent phase lags much smaller than a full oscillatory cycle are suggestive of directional influences, they are in principle ambiguous because of the cyclic nature of the signals. We measured the GPDC [131] value among these four generated sources to measure the directed connections. It can measure causality by predicting one signal from past values of another signal in terms of the degree (GPDC value). This method based on a type of P-order Multivariate Autoregressive (MVAR) model:

$$X(t) = \sum_{p=1}^P A_p(n)X(t-p) + e(t), \quad (3.3)$$

where A_p is the autoregressive coefficient matrix with the size of 4×4 and p is time lag, P is the maximum number of lags (model order), $X(t)$ is the concatenated matrix of four source signals at time t , and $e(t)$ is the residual error vector. The MVAR model order P can be calculated by evaluating and where M is the number of time series, P is the optimal model order, N is the time point and σ is the covariance matrix. The MVAR coefficients can be obtained by two different ways [126]: **1)** the mean coefficients of all single-trial MVAR coefficients, and **2)** the MVAR coefficients of the data concatenated from all single-trial source waveforms. We selected the second way to calculate the MVAR coefficients, and set each sliding time window as 2000 ms with 50 ms step between successive windows during different trails and tasks conditions. According to our previous study [126], we employed Kalman smoother method [135] to figure out the optimal estimator for MVAR coefficients, which only can rely on previous measurements and inevitable time lag.

The fitted MVAR parameters were then transformed from the time domain into the frequency domain:

$$\Lambda_{l,m}(f, t) = I - \sum_{p=1}^P A_p(t)e^{-j2\pi fp/F_s}, \quad (3.4)$$

where I is the $p \times p$ identity matrix, with the sampling rate F_s in terms of $(l \rightarrow m)_{th}$

entry, and $\Lambda_{l,m}(f, t)$ were evaluated from 1 ~ 30Hz at every 1 Hz step. The value of **GPDC** then indicating the directional connections among these four sources is calculated as:

$$GPDC_{l \rightarrow m}(f, t) = \frac{|\Lambda_{l,m}(f, t)|}{\sqrt{\sum_{m=1}^M |\Lambda_{l,m}(f, t)|^2}}, l = 1, \dots, M, \quad m = 1, \dots, M, \quad (3.5)$$

where $\Lambda_{l,m}(f, t)$ is the variance of the prediction error for order P . After the calculation of **GPDC**, the two sample t-test was used again to identify the significant time-frequency domain between baseline (0-back) and 2-back. Although 1000 times of bootstrap re-sample method was employed again and scattered significant areas were drawn with gray band (95% confidence interval level), we still sorted out the significant area through 5×5 median filter, and pick out some obvious time-frequency domains. The bootstrap method can detect the time-frequency regions, where the **GPDC** values in 2-back tasks are significantly different compared to those values in 0-back tasks. To address the problem of multiple comparisons, the significance level (p value) was corrected using a **False Discovery Rate (FDR)** procedure.

3.4 Study Results

3.4.1 Behavioral Results

We recorded participants' behavioral performance during the implementation of the tasks. As shown in Table I, both response accuracy ($p < 0.001$) and reaction times ($p < 0.001$) significantly differed between the experimental groups.

3.4.2 Scalp Topography Performance

After averaging the waveforms across subjects, we conducted a contrast between the 2-back and 0-back (baseline) conditions. Four distinct peaks are evident in Figure 3.2. The initial peaks appear at 158 ms and 324 ms, marking a shift in scalp topographic activity from the left temporo-occipital lobe to the centroparietal lobe. Notably, prefrontal hyperactivity is observed between 844 ms and 1328 ms, indicating a reorganization of activated regions towards major frontal areas. Further analysis reveals a reduction in frontal potential from 848 ms to 1328 ms, concurrent with activation of the prefrontal, frontal, and temporal lobes.

3.4.3 Band-Specific Synchrony Analysis

We conducted a detailed examination of phase-locking synchrony among the four sources illustrated in Figure 3.4. Prior to 700 ms, as shown in Figure 3.4a and 3.4b, the connection between S2 and S3 exhibited highly synchronized coherence, with the left PPC lagging behind the right PFC (mean relative phase = -17.20° , $p < 0.001$, $r = 0.943$, bootstrap test versus zero phase lag; Figure 3.4a, middle panel). This synchronization was most prominent in the late theta and early alpha bands ($6 \sim 11$ Hz). Additionally, strong phase coherence was observed for the posterior connection (mean relative phase = -4.21° , $p < 0.001$, $r = 0.875$, bootstrap test versus zero phase lag; Figure 3.4a, right panel) in the late beta band ($28 \sim 29$ Hz). In the post-700 ms phase-locked activities, as shown in Figure 3.4c and 3.4d, notable connections include the front connection between S1 and S2 (mean relative phase = -17.91° , $p < 0.001$, $r = 0.833$, bootstrap test versus zero phase lag; Figure 3.4c, upper right panel) during the late alpha and early beta bands ($11 \sim 16$ Hz), the left lateral connection between S1 and S3 (mean relative phase = 11.08° , $p < 0.001$, $r = 0.946$, bootstrap test versus zero phase lag; Figure 3.4c, lower left panel) in the middle beta band ($17 \sim 22$ Hz), and the right lateral connection between S2 and S4 (mean relative phase = 14.89° , $p < 0.001$, $r = 0.790$, bootstrap test versus zero phase lag; Figure 3.4c, lower right panel) in the early and middle beta bands ($14 \sim 19$ Hz, and $21 \sim 26$ Hz).

3.4.4 Band-Specific Directionality Analysis

Directed coherence, reflecting the direction of potential causal influence, was observed across all frequency bands from theta to beta. Figure 3.5a presents time-frequency regions exhibiting significantly increased GPDC. These significant time-frequency domains are shown in Figure 3.5a, with directed connections for different neurocognitive processes depicted in Figure 3.5b. Comparisons between 2-back and 0-back tasks revealed significant connections at several latency intervals. Notably, connection E (150–300 ms) and connection D (550–700 ms) were detected prior to the response phase, while connections A and F (700–900 ms), C (900–1100 ms), and H, B, and G (1300–1600 ms) were observed post-response. No significant differences were found between 0-back and 2-back tasks after 1600 ms, and due to the 2000 ms duration of the final procedure, this period was excluded from the analysis.

3.4.5 Neurocognitive Architecture and Component Processes of Working Memory

Building on recent research utilizing [fMRI](#) and electrophysiological methods [98], [101], Figure 3.6 illustrates the involvement of key cognitive components. Selective attention is observed during the P300 phase [5], [136], verbal rehearsal processes are evident [137], sustained activity is observed [138], and retrieval/readout processes are engaged [101], [139]. Additionally, pattern recognition [138], memory update and storage [140], and lateral inhibition [98], [101] contribute to the cognitive landscape. Preceding responses, posterior connections are crucial for selective attention. Bilateral prefrontal regions engage in verbal rehearsal and retrieval processes, while sustained attention and pattern recognition occur between the right prefrontal and left parietal regions during the 500–700 ms interval, following an initial silent period of approximately 250 ms. After the response, sustained attention and lateral inhibition unfold in the anteroposterior right hemisphere. Memory updating and encoding processes occur in bilateral prefrontal regions during the 700–900 ms interval. Between 900 and 1100 ms, cognitive and memory components are repeated to maintain brain activity during visual [WM](#) tasks. From 1100 to 1600 ms, sustained attention monitors targeted objects, and lateral inhibition mitigates the risk of failure. Lastly, we propose a novel neurocognitive architecture for [WM](#) processing in Figure 3.7, addressing current gaps in the understanding of [WM](#).

3.5 Discussion

In the present study, we employed the traditional visual n-back paradigm and two coherence methods to construct brain network models during [WM](#) tasks. These methods adaptively identified four brain sources, predominantly located in the bilateral [PFC](#) and [PPC](#), both of which are regions linked to core [WM](#) functions. Specifically, when comparing the 2-back and 0-back conditions, [PLV](#) revealed undirected brain network connections, while [GPDC](#) provided insight into directional interactions. Notably, both coherence methods yielded similar network structures, supporting the reliability of the connectivity patterns observed. Based on these findings, we propose a comprehensive model of [WM](#) that integrates unique directional cognitive and executive connections, alongside two distinct cycles for cognitive processing and memory maintenance.

Before task responses, the initial targeted stimulus triggered selective attention in the parietal regions and was subsequently encoded in visual cortex areas. The beta-band posterior connections shown in Figure 3.4a, along with the broad beta-band directional causality observed in Figure 3.5b-I, suggested that attention was aroused upon visual

fixation of the target. Contrary to our initial expectations, beta oscillations, particularly in relation to selective attentional control, appear to play a pivotal role in governing both attention and top-down processing [141]. In line with Eriksson’s framework of a core fronto-parietal circuit sustaining attention and supporting rehearsal [101], our findings combine the primary alpha coherence in Figure 3.4a with the beta directional connections in Figure 3.5b (D), suggesting that rehearsal processes are simulated between the right PFC and left PPC. Despite using different frequency bands for coherence and causality analysis, these results point to the brain’s capacity to simulate early-stage reasoning processes in a top-down fashion, linking attention with memory encoding and rehearsal networks.

3.5.1 The Maintenance Loop During WM

Frontal brain regions, especially during delay periods, have been shown to play a pivotal role in supporting sustained brain activity [142]. Previous studies suggest that sustained frontal activation during WM tasks is linked to selective processes rather than the actual encoding of memory content [143]. Meta-analyses further indicate that the left PFC, particularly its ventral region, is more engaged in non-spatial WM tasks, while the right PFC is implicated in spatial WM [105]. Our n-back task findings, particularly the directional connection D in Figure 3.5b-I, suggest a flow of information from the right PFC to the left PPC, implying that the right PFC serves as a buffer to store information for subsequent retrieval and comparison during WM tasks.

For WM to function effectively, short-term memory maintenance is essential for supporting sustained brain activity throughout the task. The interval from 300 ms to 550 ms, marked by a relative absence of significant brain activity, suggests a stable state of activation during WM performance. This “silent period” might resemble the P300 component, which is typically associated with preparatory or anticipatory processes in WM [144]. Although we did not specifically investigate the mechanisms underlying this silent period, we hypothesize that it serves as a preparatory phase that primes higher-level WM processes. Following this period, sustained top-down influences may transform the representations stored in WM to guide decision-making [122].

The red loop in Figure 3.7 highlights the memory maintenance cycle following responses, suggesting that the bilateral PFC plays a critical role in reinforcing memory information or sustaining relevant brain activity. Previous fMRI studies have shown that older adults exhibit reduced Blood-Oxygen-Level-Dependent (BOLD) signal increases in the DLPFC during memory maintenance, underscoring the importance of both manipulation and sustained attention for effective WM performance [145]. The yellow loop represents the repetitive processes that enhance short-term memory, supporting the com-

parison, correction, and updating of memory representations.

3.5.2 The Inhibition Loop During WM

Recent theories propose that WM relies not only on recurrent excitatory interactions among pyramidal neurons to sustain activity during delays, but also on lateral inhibition to modulate interneuron activity and filter out distractions [146]–[148]. This lateral inhibition mechanism ensures that unnecessary or distracting information is suppressed, enabling efficient cognitive processing during WM tasks. In the context of filtering distractors, lateral inhibition becomes hyperpolarized when background noise threatens neuronal firing [149]. Lateral inhibition is particularly critical for filtering irrelevant inputs and maintaining focus on the task at hand, as shown by the observed effects of inhibition in the red and yellow loops of Figure 3.7, which align with theories positing that attention and WM are constrained by flexible cortical connections that balance inhibitory and excitatory influences [150].

Instances of forgetting in WM may arise from insufficient lateral inhibition, leading to a reduction in neural firing and, consequently, a failure to maintain memory representations. Our findings provide preliminary evidence of prefrontal lateral inhibition in WM, especially following the delay period, to preserve memory content and prevent distraction. However, we also speculate that inadequate inhibition may lead to cognitive “stagnation,” where the brain fails to process or update information effectively. Disinhibition, on the other hand, plays a critical role in initiating subsequent brain activity, particularly as the delay period extends. This process may explain how disinhibition acts as a switch to activate subsequent cognitive processes and reactivate inhibitory networks, facilitating the continuation of WM tasks [149].

3.5.3 Conclusion and Future Directions

In this study, we have utilized phase-locking and directional coherence analyses to explore the brain network dynamics underlying WM. Our findings suggest that the bilateral PFC and PPC are key regions involved in attention, rehearsal, memory maintenance, and inhibition. We propose a network model that integrates these processes, highlighting the essential role of disinhibition in facilitating the transition between different stages of WM. Key conclusions include: (i) the bilateral PFC and PPC are crucial for maintaining WM tasks; (ii) the right PFC supports the left PFC in high-capacity WM performance; and (iii) disinhibition serves as a necessary mechanism to unlock subsequent cognitive processes after inhibition. Future work will focus on identifying abnormal network connections in WM among individuals with depression and exploring how these abnormalities affect

cognitive functioning.

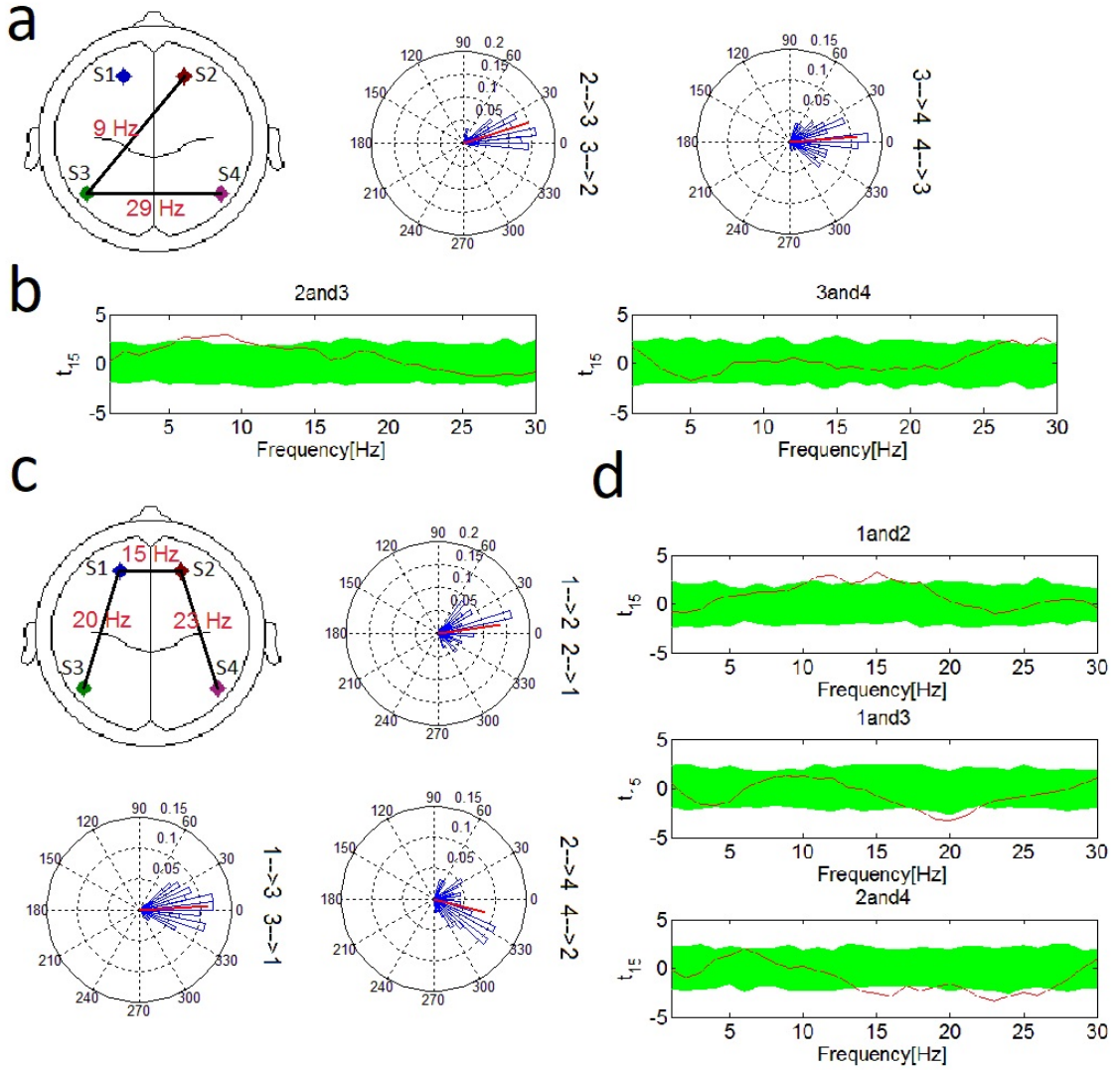


Figure 3.4: Phase-locked connections among four sources from 0 ms to 700 ms (**a**, **b**) and from 700 ms to 1600 ms (**c**, **d**). (**a**) Left panel shows connections at specific frequencies, with the right panel displaying circular statistical angles and their distribution. Circular histograms also illustrate the mean angles of phase differences between pairs of sources (red line). (**b**, **d**) t-statistics for the differences in PLV between 2-back and 0-back tasks across subjects. For example, in the pair of S1 and S3, the PLV in the 18-21 Hz beta band was higher during the 2-back task, peaking at 20 Hz. The green band represents the t-values for a one-sample t-test with a 95% confidence interval using a bootstrap method, and the red line represents the t-value. (**c**) Connections at specific frequencies with their circular statistical angles and distribution.

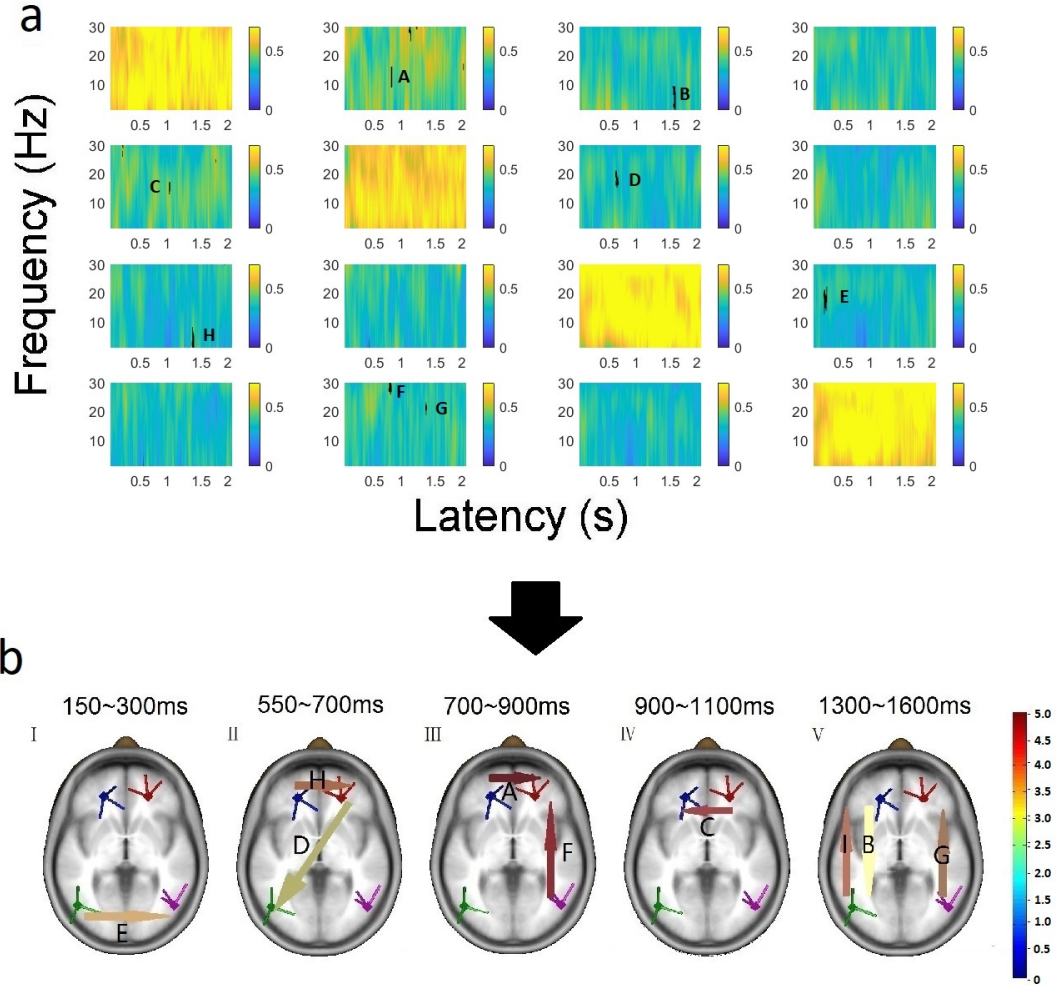


Figure 3.5: Directed connections based on the time-varying GPDC. **(a)** Time-frequency representations of the time-varying GPDC under the 2-back task, with significant grey blocks indicating differences between the 0-back and 2-back tasks using a two-sample t-test. The bar represents the GPDC value. **(b)** Directed connections at different latencies, indicated by color-coded arrows representing the direction and strength of information flow. Early latency intervals (I: 150–300 ms, E; II: 550–700 ms, D) primarily involve S3→S4 (E) and S2→S3 (D), both reflecting trigger information transmission. Late latency intervals (III: 700–900 ms, A; IV: 900–1100 ms, C; V: 1300–1600 ms, B, G, H) show diverse connections between sources reflecting memory encoding, updating, and sustained attention processes.

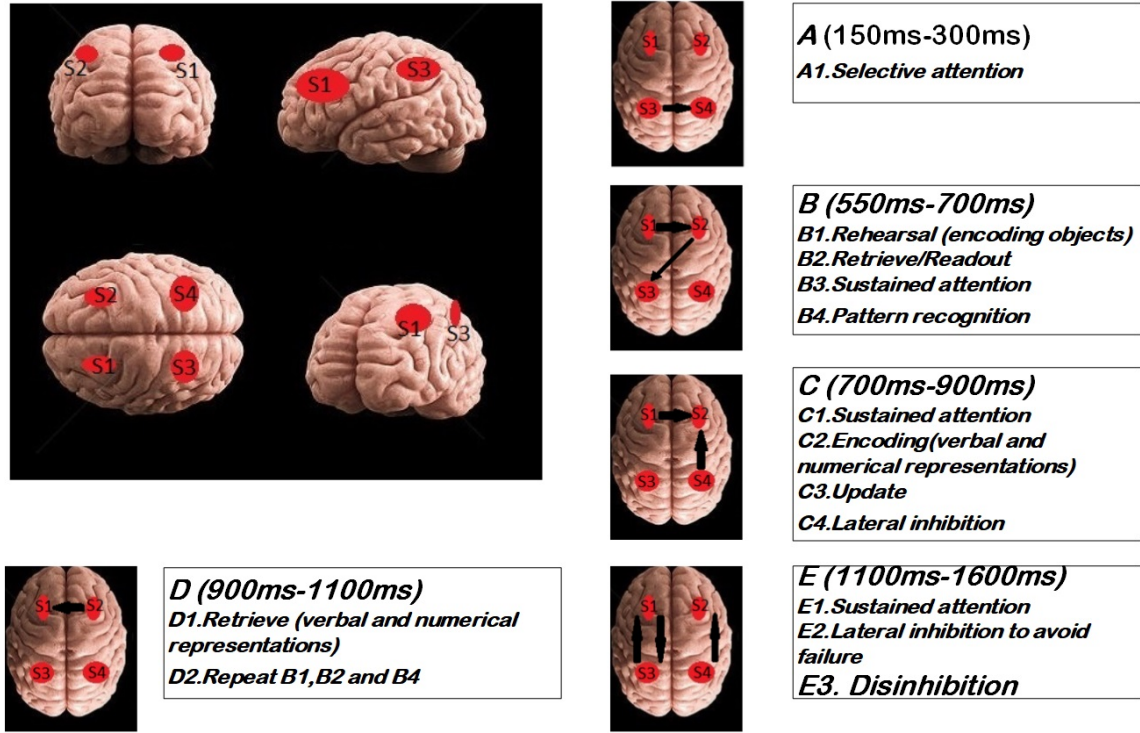


Figure 3.6: Schematic explanation of representations to brain networks during WM tasks. Left upper panel is the location illustration of four fitted sources. **A~E** present components relative to WM in terms of some specific neurocognitive processes. **A.** During this duration, selective attention is activated by the trigger of capitals shown on the screen, and this induced the attention mechanism in PPC cortex. **B.** Executive and cognitive functions between right PFC cortex and left PPC region, appear after selective attention being implemented to process numerical and verbal information. **C.** The PFC and right hemisphere connections indicate the update of information flow for memory storing, and lateral inhibition to avoid the failure of memory representation. **D.** Persistence of information under WM tasks happens in PFC cortex. **E.** The last process for the recall of sustained attention, lateral inhibition to avoid the failure of attention and memory processing, as well as disinhibition.

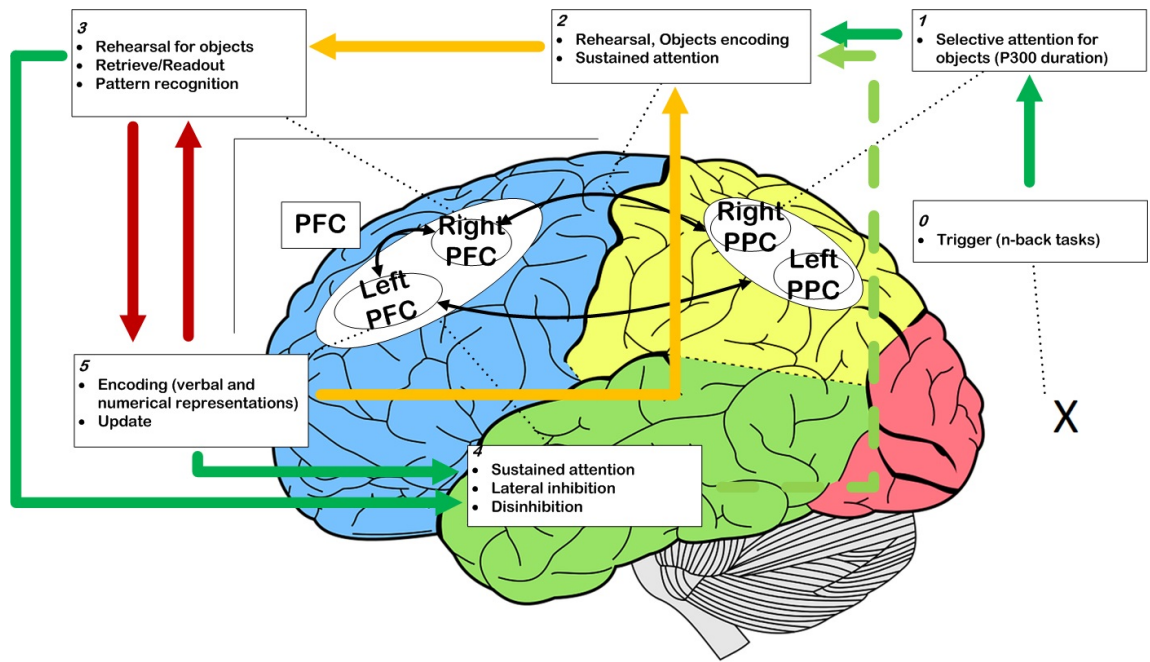


Figure 3.7: Summary of the proposed neurocognitive architecture for WM. **X** represents the visual n-back task trigger. Before responses, attention arousal (**0-1-2**) is linked with the activity maintenance loop (**2-3-2**). After the response, the brain enters a memory maintenance loop, consisting of an activity loop (**2-3-5-2**) and a major memory loop (**3-5-3**), alongside inhibition or disinhibition loops (**2-3-4-2**, **2-3-5-4-2**). The central role of inhibition is crucial for maintaining accuracy in information processing, while disinhibition resets brain activity, enabling subsequent cognitive processes.

Chapter 4

Inhibition Adaption On Pre-trained LMs

Fine-tuning pre-trained LMs may not always be the most efficient approach for downstream tasks. While traditional fine-tuning methods have shown promising results, there remains a need for a clearer understanding of their mechanisms and more effective methods for inhibiting irrelevant information. To address these challenges, we propose a novel InA fine-tuning approach that minimizes the number of tunable parameters and selectively reweights the knowledge derived from pre-trained LMs. The InA method involves two key steps: (1) inserting a small, trainable vector into each Transformer attention layer, and (2) setting a threshold to discard irrelevant knowledge. This method is inspired by the concept of shunting inhibition, where the activation of specific neurons is inhibited to regulate the flow of information in other neurons. With this inhibition mechanism, InA achieves competitive, and in some cases superior, performance compared to other fine-tuning techniques on models such as *BERT-large*, *RoBERTa-large*, and *DeBERTa-large* for tasks like text classification and question answering.

4.1 Introduction

Fine-tuning, the process of updating the parameters of pre-trained LMs, has been widely adopted as an effective approach for various downstream NLP tasks. However, classical fine-tuning methods face challenges due to the redundancy of parameters in fully pre-trained models, which can lead to inefficiencies when adapting to new tasks. To mitigate this, prior studies have attempted to adapt only specific vectors or learn additional parameters while keeping most of the pre-trained parameters fixed. This approach improves operational efficiency by allowing for the loading of task-specific parameters before model deployment. LoRA [48] has been a successful strategy, addressing both the model

depth and sequence length limitations by introducing rank decomposition to compress and reweight pre-trained parameters, achieving a balance between efficiency and performance [47], [49], [50].

However, fine-tuning pre-trained LMs for NLU tasks still faces key challenges: reducing the number of tuned weights while effectively approximating the update of pre-trained weights. Properly selecting relevant knowledge and eliminating task-irrelevant information from pre-trained models is critical. This brings us to the question: Why not directly inhibit "redundant" knowledge during fine-tuning, while preserving relevant information?

In this study, we address these challenges by proposing a novel method for fine-tuning called InA. Inspired by the shunting inhibition mechanism in neuroscience, InA offers a mechanism to suppress irrelevant knowledge during the fine-tuning process. We first provide an overview of existing adaptation fine-tuning methods and Transformer-based models, followed by a detailed explanation of activation functions, particularly focusing on the inhibition mechanism. We evaluate the performance of InA on diverse downstream tasks such as text classification, question answering, and adversarial text generation using models like Bidirectional Encoder Representations from Transformers (BERT), Robustly Optimized BERT (RoBERTa), and Decoding-enhanced BERT with Disentangled Attention (DeBERTa). In terms of storage, InA uses the same number of tunable parameters as LoRA, ensuring efficient fine-tuning while offering superior feature compression and inhibition capabilities.

Drawing on the efficiency of neural networks demonstrated in [151], and the low "intrinsic rank" concept introduced by LoRA [48], we propose InA. The key idea behind InA is to partially inhibit the intrinsic rank, thereby eliminating the influence of irrelevant "intrinsic parts" of the model. As shown in Figure 4.1, InA is similar to LoRA, as both methods optimize rank decomposition matrices while keeping pre-trained weights frozen. However, InA introduces an additional threshold mechanism to control the flow of information, inhibiting irrelevant parts. This approach allows the model to focus on task-relevant information while suppressing the influence of extraneous knowledge.

In Figure 1.3, we present a practical example of InA eliminating irrelevant knowledge from the intrinsic rank during fine-tuning. The intrinsic rank is hypothesized to follow a Gaussian-like distribution, with a concentrated center and sparse tails. By subtracting a threshold, InA removes one tail of the distribution, thereby reducing the influence of task-irrelevant features during fine-tuning. This process helps the model focus on the most relevant information for the given task.

The contributions of InA are as follows:

- (a) InA effectively inhibits irrelevant information during fine-tuning, improving model focus on task-related features and eliminating noise from task-irrelevant

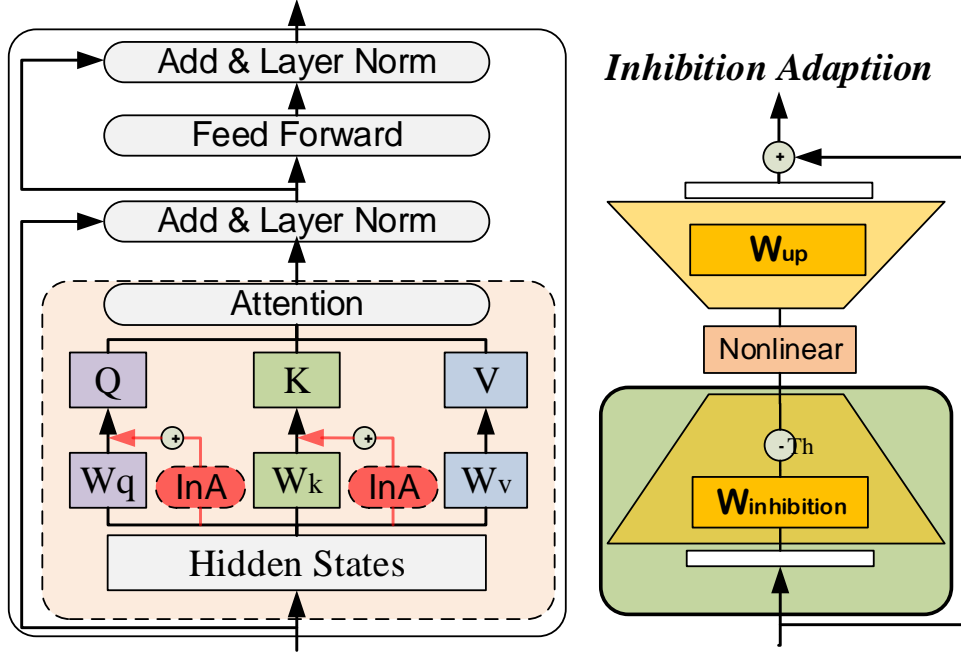


Figure 4.1: Illustration of the transformer architecture and our proposed parameter-efficient tuning method: Inhibition Adaptation.

knowledge.

- (b) **InA** benefits from activation functions with relatively flat negative tails, such as **Gaussian Error Linear Unit (GeLU)** or **Leaky Rectified Linear Unit (LeakyReLU)**, which outperform activation functions like **Rectified Linear Unit (ReLU)**. **Scaled Exponential Linear Unit (SELU)** and **Exponential Linear Unit (ELU)**, with their long and upturned tails, perform less effectively with **InA**.
- (c) **InA** shares the same trainable parameter count as **LoRA**, allowing it to inherit the knowledge compression ability of **LoRA** while adding the capability to suppress task-irrelevant knowledge through threshold-based inhibition.

4.2 Problem Statement

Previous work on **LoRA** [48] primarily focused on comparing fine-tuned models with fully fine-tuned models using similarity matrices. However, there is no direct visualization of the specific parts of the model that have been fine-tuned. Additionally, when applying **LoRA** to **LMS**, we found that although the low-rank "bottleneck" compresses information and reweights pre-trained parameters, it often introduces noise and task-irrelevant knowledge. For example, in the input sentence "I put my red bag in the black bag. What is the color of my bag?" with the target answer "red," traditional fine-tuning and **LoRA** methods

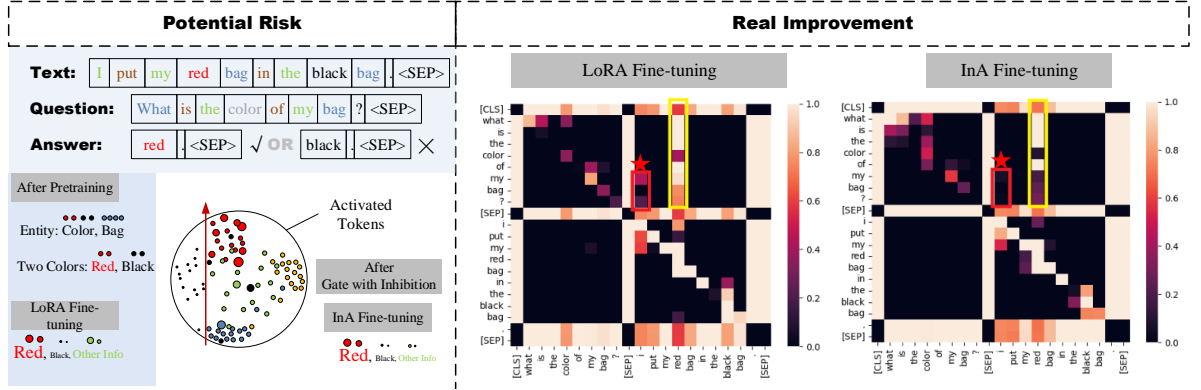


Figure 4.2: A practical example of InA and its use in the $BERT_{large}$ model, which has been fine-tuned under question-answering datasets.

struggle to eliminate irrelevant features like pronouns ("I," "my") and nouns ("bag"), which distract the model from focusing on the actual target knowledge (i.e., "red").

When fine-tuning with InA, the threshold mechanism selectively suppresses these task-irrelevant features. As shown in Figure 4.2, the inhibition vector in InA removes the influence of extraneous words, such as the pronoun "I," allowing the attention layers to focus on the most relevant terms, like "red." This process improves the model's accuracy and efficiency in answering the question.

4.3 Explanation of Shunting Inhibition

4.3.1 Shunting Inhibition (Gate with Inhibition)

The design of a gated structure with inhibition in InA is inspired by the shunting inhibition mechanism [2], [152], [153]. As shown in Figure 4.3, the gate is either "on" (red box) or "off" (green box). When the gate is off, signal transmission occurs across the joint, influenced by shunting synapses. Shunting inhibition plays a crucial role in regulating neuronal function, acting as a gating mechanism that selects, weakens, or strengthens features in the model.

In ANNs, shunting inhibition is often described as a gating mechanism, although its inhibitory function has been overlooked in some studies. Inhibition can be either subtractive, reducing membrane potential, or divisive, modulating the effect of excitation. For example, GABA receptors have both fast and slow effects on neuronal firing, which can modulate the postsynaptic potential.

4.3.2 Membrane Potentials and Threshold

As illustrated in the right panel of Figure 4.3, the threshold for inhibition is set between 10% and 30%. The membrane potentials typically range from -70mV to $+30\text{mV}$, and the threshold is set at approximately 15%. Features with activation values below this threshold are considered irrelevant and are suppressed. This mechanism helps prevent the model from focusing on unimportant features, which have little significance for specific tasks.

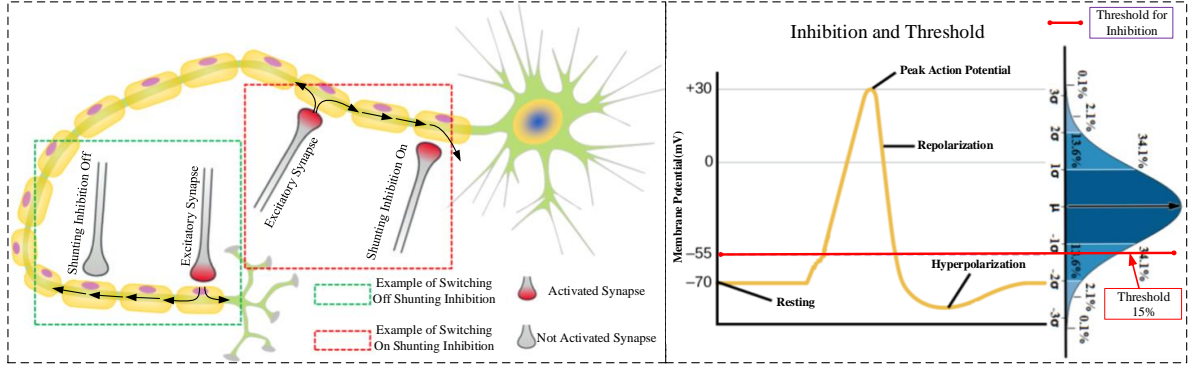


Figure 4.3: Inspiration from Neuroscience: Gate with Inhibition.

4.4 Related Work

4.4.1 Transformer-based Language Models

The Transformer ([154]), a sequence-to-sequence architecture heavily reliant on the self-attention mechanism, has revolutionized the field of NLP and achieved SOTA performance across various tasks. The exploration of Transformer scaling—by increasing model size, dataset size, model architecture, context length, and batch size—has been guided by the scaling law ([155]), significantly improving the capacity of numerous language models, such as BERT ([156]–[159]), RoBERTa ([160]), A Lite BERT (ALBERT) ([161]), DeBERTa ([162], [163]), sparse Switch-Transformer-1.6T ([164]), and Swin-Transformer ([165], [166]). Over the years, the performance of these models has improved by orders of magnitude.

For example, the self-attention operation with bias in single-head attention can be

described as ([154], [159], [162], [165]):

$$Q = HW_q + b_q, \quad K = HW_k + b_k, \quad V = HW_v + b_v, \quad (4.1)$$

$$A = \frac{QK^T}{\sqrt{d}}, \quad (4.2)$$

$$H_o = \text{softmax}(A + b_a)V, \quad (4.3)$$

where $H \in \mathbb{R}^{M \times d}$ represents the input hidden vectors, $H_o \in \mathbb{R}^{M \times d}$ is the output of the self-attention, and Q , K , and V are the query, key, and value matrices, respectively. W_q , W_k , and $W_v \in \mathbb{R}^{d \times d}$ are the projection matrices, while $A \in \mathbb{R}^{M \times M}$ is the attention matrix, and b_q , b_k , b_v , and $b_a \in \mathbb{R}^{M \times M}$ represent the bias terms. Here, M is the input sequence length and d is the dimension of the hidden states.

4.4.2 Fine-tuning on NLP Downstream Tasks

SOTA systems for **NLP** tasks largely rely on the fine-tuning of pre-trained **LMS**. In traditional fine-tuning, the pre-trained model, initially trained on a general domain, is adapted to a specific downstream task ([156]). To maximize performance, variants of the vanilla Transformer (e.g., freezing some parameters or learning only a subset of them) have been developed. This approach requires retraining all parameters of the **LM**. Fine-tuning has become the dominant paradigm for various conditional **NLP** tasks such as question answering and dialogue generation. In this paper, we focus on fine-tuning for tasks such as text classification, question answering, and text adversarial generation. We also consider three widely used pre-trained **LMS**: **BERT**, **RoBERTa**, and **DeBERTa**. However, the large size of these models poses a significant challenge for fine-tuning, as they require substantial computational resources, limiting the accessibility for practitioners.

4.4.3 Parameter-Efficient Fine-Tuning

Adapter Tuning. The adapter tuning approach inserts small, trainable modules (adapters) between Transformer layers ([46]). Each adapter uses two projection matrices, $W_{\text{down}} \in \mathbb{R}^{d \times k}$ and $W_{\text{up}} \in \mathbb{R}^{k \times d}$, to project the hidden state into a lower-dimensional space of dimension k (the bottleneck dimension). The output of the adapter is then projected back into the original hidden space. The final output after adapter tuning is given by:

$$H_o \leftarrow H_o + f(HW_{\text{down}})W_{\text{up}}, \quad (4.4)$$

where $f(\cdot)$ is a non-linear activation function. One more efficient adapter variant [167] has been proposed, and it is inserted a **Feed-Forward Network (FFN)** only after the "add

and layer norm” sub-layer.

Prefix and Infix Tuning. Prefix tuning prepends l tunable vectors to the keys and values of the multi-head attention at each layer ([47]). By concatenating or inserting two prefix vectors, $P_k \in \mathbb{R}^{M \times p}$ and $P_v \in \mathbb{R}^{M \times p}$, with the original key and value projection matrices, the new prefixed or infixed keys and values in the attention mechanism are:

$$W_k^{(i)} : \text{prefix} = \text{concat}(P_k^{(i)}, CW_k^{(i)}), \quad (4.5)$$

$$W_v^{(i)} : \text{prefix} = \text{concat}(P_v^{(i)}, CW_v^{(i)}), \quad (4.6)$$

$$W_k^{(i)} : \text{infix} = \text{insert}(CW_k^{(i)}, I_k^{(i)}), \quad (4.7)$$

$$W_v^{(i)} : \text{infix} = \text{insert}(CW_v^{(i)}, I_v^{(i)}), \quad (4.8)$$

where $C \in \mathbb{R}^{M \times d}$ represents the hidden state to be processed by the attention mechanism, and the prefix (or infix) vectors are split across multiple heads. The notation $P_k^{(i)}$, $P_v^{(i)}$, $I_k^{(i)}$, and $I_v^{(i)}$ refers to the i -th head-specific vectors.

LoRA Tuning. LoRA introduces low-rank matrices into Transformer layers to approximate the weight updates ([48]). This is achieved by decomposing the weight updates ΔW into a low-rank factorization $W_0 + \Delta W = W_0 + W_{\text{down}}W_{\text{up}}$, where $W_{\text{down}} \in \mathbb{R}^{d \times r}$ and $W_{\text{up}} \in \mathbb{R}^{r \times d}$ represent the low-rank matrices. LoRA modifies the query and value projection matrices (W_q , W_v) in the multi-head attention mechanism. The activation function used is typically LeakyReLU, with a hyperparameter value of 1.0. For a given hidden input H , the projection output is modified as follows:

$$H_o \leftarrow H_o + s \cdot f(HW_{\text{down}})W_{\text{up}}, \quad (4.9)$$

where $s \geq 1$ is a tunable scalar hyperparameter.

Other Methods. Additional parameter-efficient tuning methods include BitFit ([168]), which fine-tunes only the bias vectors, Diff-Pruning ([169]), which learns sparse parameter updates, Generalized LoRA (GLoRA) ([170]), which generalizes LoRA, and Quantized LoRA (QLoRA) ([171]), which applies quantization to LoRA using 4 or 8 bits.

4.4.4 Threshold and Inhibition

The threshold mechanism has been mostly used in deep Spike Neural Networks (SNNs) ([151], [172]). A higher threshold will prevent the neuron from firing (‘dead-neuron’ problem), and a lower threshold will cause excessive firing. Both affect the ability of the neuron to differentiate between these two input patterns ([173]). The firing thresholds are also fixed ([174]) or selected based on some heuristics ([172], [175]). The threshold was selected as the maximum preactivation of each layer in [172]. [175] selected a certain

percentile of the preactivation distribution as the threshold. Some recent works employ leak/threshold optimisation, but their application is limited to simple datasets ([176]). Most of these articles applied a threshold to [SNNs](#), but they are facing the challenge of proposing improper methods of selecting the membrane leak and the threshold. To our best knowledge, there is no example of applying inhibition to a Transformer architecture.

4.5 Inhibition Adaptation

The aim of inhibition adaption is to modify the tunable parameters by using shunting inhibition mechanism on attention blocks of Transformer. This study mainly investigated the passing information from the previous layer when using adaption fine-tuning methods. Passing through the "bottleneck", the compressed information still contain task-irrelevant content. By integrating the shunting inhibition with the adaption fine-tuning methods, the passing information not only can be compressed, but also they are able to be filtered by using the scalable threshold. Some special tokens, such as *SEP* and *CLS*, are also fine-tuned accordingly, even through theses special tokens are important to BERT-based [LMs](#). BERT-based language models considered two directions [156]. [Generative Pre-trained Transformer \(GPT\)](#)-based models will consider two directions during the pre-training procedure, but during the fine-tuning and inference, they only consider the direction from the left to the right [157]. Given a threshold, [InA](#) is designed to fine-tune the Transformer block, and it aims to inhibit the attention block accordingly. Therefore, when fine-tuning on [BERT](#)-based [LMs](#) by using [InA](#), [InA](#) will consider two directions. In contrast, when fine-tuning on [GPT](#)-based [LMs](#) via [InA](#), [InA](#) only consider the direction of left-to-right.

4.5.1 Inhibited Adaptation

InA introduces trainable inhibition matrices into Transformer layers to approximate weight updates. By utilizing a low-rank decomposition $W_0 + \Delta W = W_0 + W_{\text{down}}W_{\text{up}}$, where $W_{\text{down}} \in \mathbb{R}^{d \times r}$, $W_{\text{up}} \in \mathbb{R}^{r \times k}$, and $Th \in \mathbb{R}^{M \times 1}$, InA updates the Query and Key projection matrices (W_q, W_k) in the multi-head attention sub-layer. For a given input H , InA modifies the projection output H_o as:

$$H_o \leftarrow H_o + s \cdot f(HW_{\text{down}} - Th)W_{\text{up}}, \quad (4.10)$$

where $s \in \{0, 1\}$ is a tunable scalar hyperparameter, and Th is the threshold.

Notation. We denote the input hidden vectors as $H \in \mathbb{R}^{M \times d}$ and the output of the self-attention mechanism as $\bar{H}_o \in \mathbb{R}^{M \times d}$. The projection matrices $W_k, W_q, W_v \in \mathbb{R}^{d \times d}$ correspond to the Key, Query, and Value components, respectively.

Motivation. The motivation behind InA in Transformers is to introduce a flexible gating mechanism, using an adjustable inhibition vector to fine-tune downstream tasks. This mechanism automatically learns to filter out irrelevant features, avoiding the need for explicit sparsity constraints. In the context of transfer learning, pre-trained language models can provide general features, while the inhibition vector with its gating mechanism learns to refine and suppress unnecessary information. This makes it possible to adjust the weights for more effective task-specific fine-tuning. We formulate the linear InA layer as:

$$I_k = f(HW_{k_down} - Th_k)W_{k_up}, \quad (4.11)$$

$$I_q = f(HW_{q_down} - Th_q)W_{q_up}, \quad (4.12)$$

where $I_k \in \mathbb{R}^{M \times d}$ and $I_q \in \mathbb{R}^{M \times d}$ represent the inhibition matrices on the Key and Query sides, respectively. Here, f is the activation function, and $Th_k \in \mathbb{R}^{M \times 1}$ and $Th_q \in \mathbb{R}^{M \times 1}$ are thresholds derived from column-wise max operations on the pre-activation values:

$$Th_k = \max(HW_{k_down}) \times \text{Inh}_p, \quad (4.13)$$

$$Th_q = \max(HW_{q_down}) \times \text{Inh}_p. \quad (4.14)$$

4.5.2 Inserting InA into Transformer

The next challenge is to adjust the adaptivity of LMs and identify the most relevant features from the extensive feature pool generated during pre-training. To achieve this, we propose subtracting a threshold (Th_q) as shown in Equations (4.13) and (4.14). This subtraction mechanism acts as a filter, allowing the model to discard features with negative activations. With the introduction of inhibition, as depicted in the right panel of Figure 1.3, irrelevant features (e.g., the extra knowledge about "I" and "my" in the red box) are suppressed. Using the Gaussian Error Linear Unit (GELU) activation function, the inhibition matrices I_k and I_q cut off the long negative tail of the activation distributions, thereby retaining the more concentrated, useful features. This selective process enhances the ability of the attention blocks to focus on dense and relevant features during fine-tuning.

Next, we incorporate InA into the Transformer attention mechanism. The linear InA modification of the Transformer is formulated as follows:

$$V = HW_v + b_v, \quad K = HW_k + b_k, \quad Q = HW_q + b_q, \quad (4.15)$$

$$B_k = K + I_k, \quad B_q = Q + I_q, \quad \bar{A}_{kq} = \frac{B_q B_k^T}{\sqrt{d}}, \quad (4.16)$$

Table 4.1: Hyper-parameters for fine-tuning BERT, RoBERTa and DeBERTa with inhibited gate MLPs mechanism on down-streaming tasks.

Hyper-parameter	BERT(large)	RoBERTa(large)	DeBERTa(large)
Dropout of task layer	0.15	0.15	0.15
Warmup Steps	100	100	100
Learning Rates	5e-6	5e-6	5e-6
Batch Size	{16,32,64}	{16,32,64}	{16,32,64}
Weight Decay	0.01	0.01	0.01
Epochs	5	10	10
Learning Rate Decay	Linear	Linear	Linear
Optimizer	AdamW	AdamW	AdamW
Adam ϵ	1e-6	1e-6	1e-6
Adam (β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Gradient Clipping	1.0	1.0	1.0
Inhibition Percentile	(0.0, 0.1, 0.3, 0.9)	(0.0, 0.1, 0.3, 0.9)	(0.0, 0.1, 0.3, 0.9)

$$\bar{H}_o = \text{softmax}(\bar{A}_{kq} + b_{\bar{a}})V, \quad (4.17)$$

where $V \in \mathbb{R}^{M \times d}$ is the Value matrix, B_k and $B_q \in \mathbb{R}^{M \times d}$ are the modified Key and Query matrices with InA, and $\bar{A}_{kq} \in \mathbb{R}^{M \times M}$ is the attention matrix. The term $b_{\bar{a}} \in \mathbb{R}^{M \times M}$ represents the relative position bias term for each head in the multi-head attention mechanism.

Equations (4.15) and (4.16) follow the same structure as the vanilla Transformer attention mechanism. They generate the Key, Query, and Value projection matrices to represent context attributes. However, the introduction of the inhibition matrices I_k and I_q in Equations (4.13) and (4.14) enables fine-grained control over the feature selection process. This allows the model to adjust the context attributes more precisely to fit the downstream task during fine-tuning. By adding I_k to K and I_q to Q , the model can retain or enhance important context features, while suppressing those deemed irrelevant or counterproductive.

4.6 Experiments

4.6.1 Experiment Settings

Our experiments only depend on single-task fine-tuning. Our code is implemented based on the Huggingface Transformer ([177]). Following prior studies of language models ([48], [169]), we report results using large models. We use $8 \times$ NVIDIA Tesla A100 with 40GB graphic memory cards to fine-tune the pre-trained models. Code and models are available at: <https://github.com/ChengKang520/inhibited-lora>.

4.6.2 Evaluation Datasets

This section evaluates the performance of **InA** in terms of downstream tasks on *BERT – large* ([161]), *RoBERTa – large* ([160]) and *DeBERTa – large* ([162], [163]). Whether natural language understanding, question answering or generation, specifically, the benchmark **General Language Understanding Evaluation (GLUE)** ([178]), **Stanford Question Answering Dataset (SQuAD)** v1.1 ([179]), **SQuAD v2.0** ([179]) and **Situations With Adversarial Generations (SWAG)** ([180]), we followed the adapter fine-tuning setup ([167]) on *RoBERTa – large* for a direct and fair comparison. Refer to Table 5.4 for detailed hyperparameters.

4.6.3 Fine-Tuning Implementation Details

Settings. Following **BERT** ([156]), **RoBERTa** ([160]) and **DeBERTa** ([162]), we adopt dynamic data batching. We also include span masking ([181]) as an additional masking strategy with a span size of up to three. For fine-tuning, we use Adam ([182]) as the optimiser for a fair comparison, and we train each task with a hyperparameter search procedure—each run takes about 1–2 hours on a DGX-2 node. All the hyperparameters are presented in Table 5.4. The model selection is based on the performance of the task-specific sets.

Our experiments are under fine-tuning on downstream tasks. Firstly, we set the inhibition percentile as 0% to test whether the result is similar to the settings without inhibited gate **MLPs**. Secondly, we set the inhibition percentile as 10% or 90% according to the performance of the first step. Finally, if the result, when the inhibition percentile is 10%, becomes better, we will set the inhibition percentile as 30%. If not, we will set the inhibition percentile as 90%.

4.7 Results

We summarise the efficiency performance of adaption **fine-tuning (FT)** methods and **InA** in Table 4.2. In addition to comparing with different adaption methods, by inserting **InA** into *BERT – large*, *RoBERTa – large* and *DeBERTa – large*, we also summarise the results on eight **NLU** tasks of **GLUE** ([178]) in Table 5.1, as well as question answering – **SQuAD** v1.1 ([179]), **SQuAD** v2.0 ([183]) and Text Adversarial Generation: **SWAG** ([180]) in Table 5.2. In Table 5.3, we compare the performance of **InA** on the **GLUE** development set when fine-tuning *BERT – large* with five epochs over five different activation functions. We also summarise the performance of different inhibition levels on these three large language models in Table 5.5. **BERT**-based language models consider two

directions during pretraining and fine-tuning. GPT-based models consider two directions during the pre-training procedure, but during the fine-tuning and inference, they only consider the direction from the left to the right.

4.7.1 Efficiency: Trainable Parameters and Speed

Additionally, we would like to discuss the efficiency gains of InA, such as the reduction in trainable parameters, and back-propagation speed and inference (complexity). We treat W_q (or W_k , W_v) as a single matrix of dimension $d \times d$. We denote the number of the prefix (resp. infix) tokens as l_p (resp. l_i). r is the low-rank mechanism that controls the bottleneck. In Table 4.2, the activation function of adapter FT is ReLU; Prefix uses Softmax (Softmax); LoRA is thought to has a LeakyReLU activation function (slope is 1.0), and InA uses LeakyReLU (default slope). Eventually, InA shows the fewest tunable parameters but the same inference complexity when using LeakyReLU. In Table 4.2, LeakyReLU has no obvious average gap with GeLU, because they almost have the same function and derivative waveform. In Table 4.2, L is the number of fine-tuned layers, $\mathcal{O}(n)$ is the computational complexity in terms of the sequence length n [154]. Multi-head attention consists of several attention layers are running in parallel [154], and LoRA can be seen as external modules added in a parallel manner [48]. Thus, the inference computation of LoRA block also should be considered. The inference and fine-tuning time of InA is longer than LoRA, as InA applies additional maximizing and subtracting implementations on attention metrics.

Table 4.2: The efficiency of InA and other adaptation FT methods in terms of trainable parameters, inference (complexity), and update speed (back-propagation).

Methods	Tunable Params	Inference	Complexity per Layer
Fully FT	$T1 = 3 \times L \times d^2$	$T1$	$\mathcal{O}(n^2)$, GeLU
Adap FT	$T2 = 2 \times L \times d \times r + r + d$	$T1 + T2$	$\mathcal{O}(n)$, ReLU
Prefix FT	$T3 = L \times d \times (l_p + l_i)$	$T1 + T3$	$\mathcal{O}(n^2)$, Softmax
LoRA FT	$T4 = 2 \times L \times d \times r$	$T1 + T4$	$\mathcal{O}(n)$, LeakyReLU
InA FT	$T5 = 2 \times L \times d \times r$	$T1 + T5$	$\mathcal{O}(n)$, LeakyReLU $\mathcal{O}(n^2)$, GeLU

4.7.2 Effectiveness: InA on Fine-tuning

Our settings for *BERT – large* and *DeBERTa – large* on InA are, respectively, similar to the input/output protocol for BERT ([156]) and DeBERTa ([163]) fine-tuning. Our settings for InA fine-tuning on *RoBERTa – large* are, respectively, similar to the adaption fine-tuning method ([48], [167]).

Table 4.3: Comparison results of fine-tuning the GLUE development set on *BERT-large*, *RoBERTa-large*, *DeBERTaV2-large* and *DeBERTaV3-large* with *InA* (inhibition level percentile is 0.3). † indicates runs configured in a setup similar to [46] for a fair comparison.

Model-large & Method #Train	#Trainable Parameters	CoLA Mcc 8.5k	QQP Acc 364k	MNLI Acc 393k	SST2 Acc 67k	STS-B Corr 7k	QNLI Acc 108k	RTE Acc 2.5k	MRPC Acc 3.7k	Avg.
BERT [156]	336.0M	60.6	91.3	86.6	93.2	90.0	92.3	70.4	88.0	84.5
BERT [FT] †	336.0M	64.0	91.3	86.2	93.8	88.9	92.6	71.4	86.6	84.35
BERT [LoRA] †	0.8M	64.2±0.7	91.4±0.2	86.2±0.2	94.2±0.2	89.2±0.2	92.7±0.1	69.2±1.4	84.9±1.3	84.01
BERT [InA] †	0.4M	65.9±0.6	91.5±0.1	86.3±0.2	94.4±0.2	89.0±0.2	92.7±0.1	69.0±1.6	84.8±1.1	84.19
RoBERTa [160]	355.0M	68.0	92.2	90.2	96.4	92.4	93.9	86.6	90.9	88.82
RoBERTa [FT] †	355.0M	68.1	92.2	90.2	96.3	92.3	93.9	86.6	90.9	88.56
RoBERTa [Adpt]†[167]	0.8M	67.8±2.5	91.7±0.2	90.5±0.3	96.6±0.2	91.9±0.4	94.8±0.3	80.1±2.9	89.7±1.2	87.9
RoBERTa [Adpt]†[46]	0.8M	66.3±2.0	91.5±0.1	90.3±0.3	96.3±0.5	91.5±0.5	94.7±0.2	72.9±2.9	87.7±1.7	86.4
RoBERTa [LoRA]†[48]	0.8M	68.2±1.9	91.6±0.2	90.6±0.2	96.2±0.5	92.3±0.5	94.8±0.3	85.2±1.1	90.2±1.0	88.6
RoBERTa [InA] †	0.4M	68.5±1.2	92.2±0.1	90.2±0.4	96.4±0.3	92.0±0.3	94.4±0.4	85.2±0.7	90.8±0.5	88.7
DeBERTaV2 [162]	304.0M	70.5	92.3	91.1	96.8	92.8	95.2	88.3	91.9	90.00
DeBERTaV3 [163]	304.0M	75.3	93.0	91.8	96.9	93.0	96.0	92.7	92.2	91.37
DeBERTaV3 [FT] †	304.0M	74.3	93.0	91.0	96.2	92.6	95.4	90.3	90.7	90.44
DeBERTaV3 [LoRA] †	0.8M	75.6±1.2	93.1±0.1	91.0±0.2	96.6±0.3	92.8±0.2	96.0±0.1	91.2±0.7	92.9±0.2	91.15
DeBERTaV3 [InA] †	0.4M	76.4±1.0	93.2±0.1	90.9±0.3	96.6±0.4	93.2±0.2	96.1±0.1	90.7±0.8	93.1±0.2	91.28

Table 4.4: Comparison results of fine-tuning SQuAD v1.1, SQuAD v2.0 and SWAG on *BERT-large*, *RoBERTa-large*, *DeBERTaV2-large* and *DeBERTaV3-large* with *InA* (inhibition level percentile is 0.9). ★ indicates being run under the original configuration for a fair comparison. (Note that missing results in the literature are signified by ‘-’).

Model-large & Method #Train	# Trainable Parameters	SQuAD v1.1 F1/EM	SQuAD v2.0 F1/EM	SWAG Acc
BERT [156]	336.0M	90.9/84.5	81.8/79.0	88.6
BERT [FT] ★	336.0M	91.3/84.5	81.7/78.4	86.5
BERT [LoRA] ★	0.8M	91.3/84.5	81.7/78.4	86.5
BERT [InA] ★	0.4M	91.3/84.6	81.5/78.1	86.7
RoBERTa [160]	355.0M	94.5/88.9	89.4/86.5	89.9
RoBERTa [FT] ★	355.0M	94.1/88.4	88.9/86.0	88.9
RoBERTa [LoRA] ★	0.8M	94.4/88.7	88.8/86.0	88.9
RoBERTa [InA] ★	0.4M	94.7/89.2	89.1/86.3	88.9
DeBERTaV2 [162]	304.0M	95.5/90.1	90.7/88.0	90.8
DeBERTaV3 [163]	304.0M	-	91.5/89.0	93.4
DeBERTaV3 [FT] ★	304.0M	95.4/89.8	91.5/89.0	93.3
DeBERTaV3 [LoRA] ★	0.8M	95.3/89.9	91.5/89.0	93.2
DeBERTaV3 [InA] ★	0.4M	95.4/90.0	91.6/89.0	93.3

4.7.3 InA on the Text Classification Task

We summarise the comparison results on these eight NLU tasks in Table 5.1 in terms of fine-tuning the architecture of inserting *InA* into the original *BERT-large* *RoBERTa-large* and *DeBERTa-large*. In Table 5.1, when using *BERT-large* as the base, the average cannot catch up with the performance of using the classical *FT* method, but *InA* fine-tuning outperforms the classical *FT* method on six out of eight tasks. Although *RoBERTa-large* with *InA* fine-tuning merely shows the fine-tuning advantage on *Corpus of Linguistic Acceptability* (CoLA), *Quora Question Pairs* (QQP) and *Microsoft Research Paraphrase Corpus* (MRPC) tasks, it can achieve the highest average result. Figure 4.6 shows the attention heatmap when using *InA* to fine-tune the GLUE tasks. Fine-tuning *DeBERTaV3-large* with *InA* on GLUE can get five out of eight better results, even

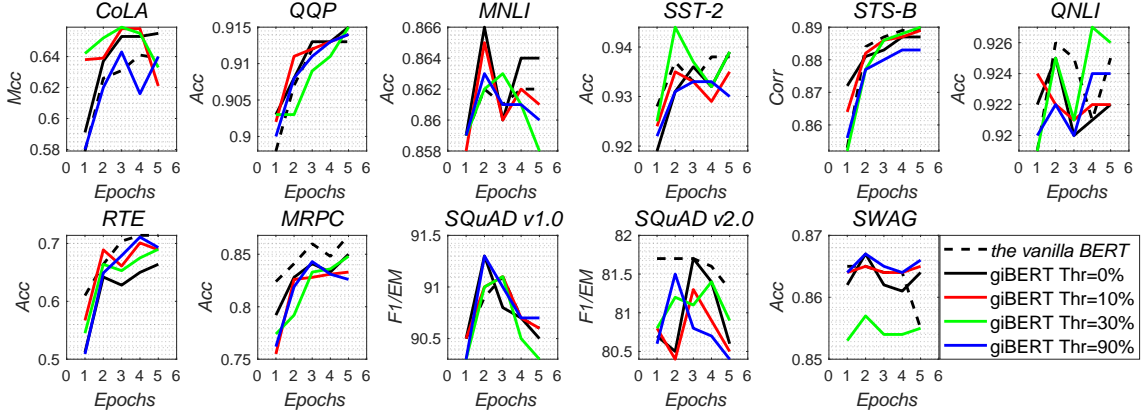


Figure 4.4: Plots of corresponding metrics according to the number of epochs on the validation split of GLUE, SQuAD v1.1, SQuAD v2.0 and SWAG. The giBERT means inserting InA (gate inhibition mechanism) into BERT.

though it also cannot achieve a better average. From Table 5.1, we can find that when fine-tuning [Recognizing Textual Entailment \(RTE\)](#) and [MRPC](#) under InA, *BERT – large* and *RoBERTa – large* cannot always get a better performance than other FT methods. The inferred reason is that the extra tunable parameters cannot be efficiently fine-tuned with small data.

InA on the Question Answering Task

As we use three large language models as the baseline, *BERT – large*, *RoBERTa – large* and *DeBERTa – large*, when fine-tuning with InA on [SQuAD v1.1](#) and [SQuAD v2.0](#) ([179]), we can find a weak improvement in Table 5.2. Moreover, the obviously dominant part is that InA inhibits the ‘irrelevant knowledge’ (e.g., ‘I’ and ‘my’) when $Inh_p = 0.9$ (See Figure 4.7). We infer that InA inhibits the information that has a relationship with the label (the label is ‘red’), for example, the word ‘my’ in the phrase ‘my red’. That is why InA can achieve relatively better results on the [SQuAD](#) task. InA is not only intended to fine-tune BERT-based LMs, and we report the visualization results on [SQuAD-V2](#) in terms of using the InA fine-tuning method on *RoBERTa – large* [160] and *Llama2* [62] (Seen Figures 4.8 and Figure 4.9).






InA on the Multi-Choice Generation Task

In Table 5.2, for the [SWAG](#) text generation dataset ([180]), which introduces the task of grounded commonsense inference, unifying natural language inference and commonsense reasoning, we find there is no fine-tuning improvement. In Figure 4.13, the input is ‘she opened the hood of the car’. Humans can reason about the situation and anticipate what might come next (the label is ‘then, she examined the engine’). The inhibitor can reduce

the influence of some information, but the reason why such ‘unimportant knowledge’ is required for the [SWAG](#) task is still not clear. We will perform more experiments to figure out the reason why [InA](#) cannot benefit [SWAG](#) in our future work.

Different Activation Functions on InA

Table 4.5: When using different activation functions, we set the inhibition level percentile at 0.3 and present the comparison results on the GLUE development set within five epochs fine-tuning based on *BERT – large*.

Model-large #Train BERT(30%)	GeLU	SELU	ELU	LeakyReLU	ReLU
Functions					
CoLa (Mcc)	65.9	62.1	62.8	66.6	64.3
QQP (Acc)	91.5	63.2	63.2	91.4	91.4
MNLI (Acc)	86.3	35.4	35.5	86.3	86.3
SST2 (Acc)	94.4	50.9	92.9	93.6	93.1
STS-B (Corr)	89.0	32.0	77.0	88.9	89.3
QNLI (Acc)	92.7	50.5	92.0	92.3	92.3
RTE (Acc)	69.0	54.9	52.7	70.0	68.6
MRPC (Acc)	84.8	68.4	77.2	84.3	83.8
Avg.	84.20	44.41	69.15	84.18	83.64

We summarise the results of using different activation functions after setting the inhibition percentile at 30% in Table 5.3. When compared with other activation functions whose tails are zero or negative, the [GeLU](#) activation function, whose negative tails are short, achieves the best improvement of [QQP](#), [Stanford Sentiment Treebank \(SST2\)](#), [Stanford Question Answering Dataset \(QNLI\)](#), [MRPC](#) and [GLUE](#) averages. Although [LeakyReLU](#) with a default slope gets outstanding performance on [CoLA](#) and [RTE](#), the total effect on [GLUE](#) tasks is inferior to [GeLU](#). [LeakyReLU](#) can provide more stable and smoother negative values, and this could be the reason why [LeakyReLU](#) can outperform [GeLU](#) on these two small downstream [GLUE](#) tasks. The negative value deriving from [LeakyReLU](#) activation would provide a stronger inhibition for [BERT](#) or variants of [BERT](#) ([RoBERTa](#), [DeBERTaV2](#) and [DeBERTaV3](#)). [GeLU](#) has a short and tender negative tail, and we eventually select it as the default activation function.

In Table 5.3, every activation function has its negative tail, except [ReLU](#). Because the inhibition vector has subtracted one inhibition variable through the [GeLU](#) and [LeakyReLU](#) activation functions, some variables become negative, and the output of the inhibition layer at the end has more negative variables if setting Inh_p higher. Thus, we can slightly ‘reweight’ the Q and K matrices with this inhibition vector. The worse performance of [SELU](#) can be a contrary example because it has an upturned tail which provides bigger negative outputs.

Inhibition Level in InA

Table 4.6: Comparison results on fine-tuning the GLUE development set, SQuAD v1.1, SQuAD v2.0, and SWAG—Inserting InA into *BERT – large*(1*), *RoBERTa – large*(2*) and *DeBERTa – large*(3*). The values after each model are inhibition levels.

Model #Train		GLUE								SQuAD v1.1	SQuAD v2.0	SWAG	
(Large Model on InA)		CoLA Mcc 8.5k	QQP Acc 364k	MNLI Acc 393k	SST2 Acc 67k	STS-B Corr 7k	QNLI Acc 108k	RTE Acc 2.5k	MRPC Acc 3.7k	Avg.	F1/EM 87.6k	F1/EM 130.3k	Acc 73.5k
1*	BERT(0)	65.5	91.5	86.6	93.9	88.7	92.5	66.4	85.0	83.76	91.1/84.3	81.6/78.9	86.6
	BERT(0.1)	65.8	91.4	86.5	93.5	88.9	92.4	70.1	83.1	83.96	91.1/84.4	81.3/78.5	86.5
	BERT(0.3)	65.9	91.5	86.3	94.4	89.0	92.7	69.0	84.8	84.19	91.1/84.4	81.4/78.1	86.7
	BERT(0.9)	64.3	91.4	86.3	93.3	88.3	92.4	71.1	84.3	83.70	91.3/84.6	81.5/78.1	86.7
2*	RoBERTa(0)	64.1	92.2	90.2	95.8	92.0	94.1	85.2	89.0	87.81	93.9/88.4	88.3/84.7	88.3
	RoBERTa(0.1)	65.5	92.0	89.5	95.6	92.4	94.4	83.4	91.7	88.05	94.1/88.8	88.5/85.5	88.4
	RoBERTa(0.3)	68.5	92.2	90.2	96.4	92.0	94.4	85.2	90.8	88.69	94.2/88.8	88.7/85.3	89.6
	RoBERTa(0.9)	67.5	92.1	89.6	95.8	91.6	94.1	85.2	89.7	88.20	94.7/89.2	89.1/86.3	89.9
3*	DeBERTaV3(0)	73.2	93.1	90.9	96.6	93.2	95.5	90.3	91.4	90.65	95.2/89.7	90.8/88.5	91.9
	DeBERTaV3(0.1)	76.5	93.2	90.8	96.2	93.2	96.0	90.0	92.3	91.03	95.3/89.9	91.2/88.7	93.3
	DeBERTaV3(0.3)	76.4	93.2	90.9	96.6	93.2	96.1	90.7	93.1	91.28	95.4/89.9	91.1/88.4	93.5
	DeBERTaV3(0.9)	72.8	93.0	90.9	96.2	92.6	95.5	89.5	90.7	90.19	95.4/90.0	91.6/89.0	93.3

We also summarise the performance of using four different inhibition levels in Table 5.5. For text classification tasks, when the inhibition level percentile is 0.3, InA can achieve the dominant results. In Figure 4.4, the inhibition mechanism affects the fine-tuning performance, especially when the inhibition level is between 10% and 30%. But for the question-answering and adversarial text-generation tasks, when the inhibition level percentile is 0.9, there is a weak improvement.

Trainable Weights by Using s on InA

InA on Single Key or Query Side. For the single side conditions (either on the *Key* or on the *Query*) and based on *DeBERTaV3 – large*, we summarise the results in Table 4.7. When the inhibition level Inh_p is 0.3, we get the best GLUE average using InA both on the *Key* and on the *Query*. There are two unexpected findings when inserting InA into the single attention side (*Key* or *Query*). The first is that when setting the inhibition level $Inh_p = 0.0$, we can achieve the best result at 92.1% in terms of fine-tuning the RTE task. The second is that when fine-tuning the downstream SQuAD v1.1 task with 0.3 and 0.1 inhibition levels, the Key side and the Query side respectively present the best result at 95.8%/89.3% and 95.8%/89.5%.

Inserting InA into Several Last Layers. To find the best inserting position, for example, which layer in BERT-like architectures needs inhibition, as well as ascertain how deep the inhibition should be set (for example, from the 16_{th} layer to the 24_{th} layer), we summarise the relevant results in Table 4.8 based on *DeBERTaV3 – large*. We roughly disassemble the DeBERTa architecture in Figure 4.5 and, depending on this, we insert InA into several last layers (last 3, 6 and 12 layers).

Table 4.7: Comparison results on fine-tuning the GLUE development set, SQuAD v1.1, SQuAD v2.0, SWAG and NER. (Note that **Key*** and **Query*** respectively mean inserting InA into Transformers’ Key side and Query side).

	Model #Train	GLUE									SQuAD v1.1	SQuAD v2.0	SWAG
		CoLA Mcc 8.5k	QQP Acc 364k	MNLI-m/mmSST2 Acc 393k	Acc 67k	STS-BQ/NLI Corr 7k	RTE Acc 108k	Acc 2.5k	MRPC Acc 3.7k	Avg.	F1/EM 87.6k	F1/EM 130.3k	Acc 73.5k
Key*	(Large)												
	giDeBERTaV3(0)	72.6	93.0	90.9/90.9	96.3	92.8	95.4	88.8	92.2	90.25	94.8/89.2	89.9/86.5	92.2
	giDeBERTaV3(0.1)	74.0	93.0	91.2/91.0	96.2	92.9	95.4	89.5	91.9	90.51	94.8/89.3	89.7/86.9	91.6
	giDeBERTaV3(0.3)	75.0	93.1	91.0/90.9	96.2	92.8	95.3	91.7	91.7	90.85	95.8/89.3	89.9/86.4	92.2
	giDeBERTaV3(0.9)	72.0	93.1	91.0/91.0	96.3	92.8	95.4	91.3	91.4	90.41	94.8/89.3	90.3/86.9	92.0
Query*	giDeBERTaV3(0)	71.9	93.0	91.0/90.9	96.2	92.8	95.3	92.1	90.2	90.31	94.7/89.2	90.1/86.9	92.2
	giDeBERTaV3(0.1)	73.2	92.9	91.3/90.9	96.3	92.7	95.1	89.2	90.2	90.11	95.8/89.5	90.4/87.7	92.2
	giDeBERTaV3(0.3)	73.5	92.9	91.3/90.9	96.2	93.0	95.4	89.5	91.9	90.46	94.8/89.3	89.7/86.9	91.6
	giDeBERTaV3(0.9)	74.2	93.0	90.8/90.8	95.6	92.9	95.4	90.6	90.2	90.34	94.8/89.5	89.8/86.7	92.0

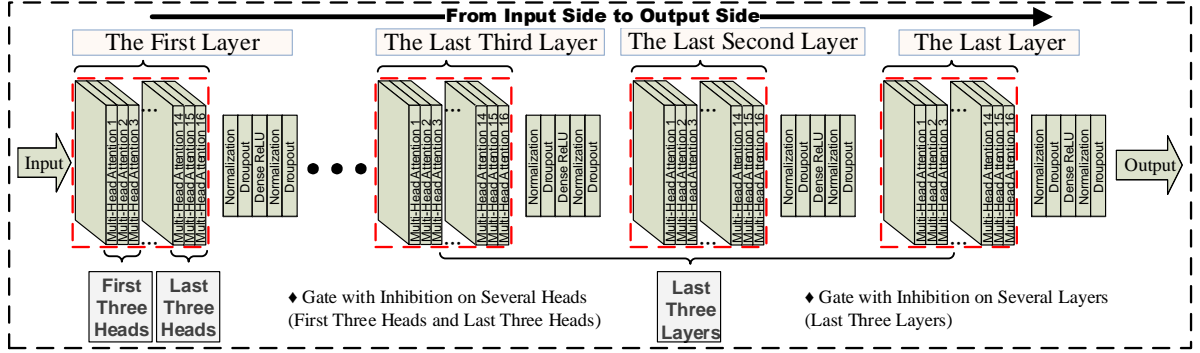


Figure 4.5: Roughly disassembled DeBERTaV3 architecture.

4.8 Analysis and Discussion

We now empirically validate the effectiveness of InA. Based on experimental results from the benchmarks, we address and answer the following three key questions:

Q1: Is inhibition necessary during adaptation fine-tuning, and how does the InA method work in this context?

Q2: If inhibition is needed, how should we choose the inhibition level Inh_p and select an appropriate rank r in practical scenarios?

Q3: Does the inhibition adaptation matrix $W_{inhibition}$ effectively inhibit irrelevant knowledge? If so, which specific irrelevant knowledge is suppressed in practice?

We believe that the answers to **Q2** and **Q3** provide valuable insights into the fundamental principles of using pre-trained language models for downstream tasks.

4.8.1 Difference Between LoRA and InA

We conducted experiments to ensure a fair comparison with LoRA. From Figure 4.6a) to Figure 4.13a), when the inhibition level is set to 0, i.e., when InA is initialized as LoRA, InA can reweight the pre-trained parameters. However, if InA is set with a higher inhibition level, such as $Inh_p = 0.3$ (Figure 4.6c) to Figure 4.13c)), InA can adaptively

Table 4.8: Comparison results on fine-tuning the GLUE development set, SQuAD v1.1, SQuAD v2.0, SWAG and NER on language models’ several last layers.

Model #Train		GLUE									SQuAD v1.1	SQuAD v2.0	SWAG
(Large Model on InA)		CoLA	QQP	MNLI	SST2	STS-B	QNLI	RTE	MRPC		F1/EM	F1/EM	Acc
		Mcc	Acc	Acc	Acc	Corr	Acc	Acc	Acc	Avg.	87.6k	130.3k	73.5k
		8.5k	364k	393k	67k	7k	108k	2.5k	3.7k				
Last 3	DeBERTaV3(0)	73.5	92.9	91.0	96.6	92.8	95.5	89.2	90.7	90.27	94.7/89.1	89.7/86.9	91.4
	DeBERTaV3(0.1)	73.2	93.0	90.9	96.5	92.9	95.8	90.6	91.1	90.50	94.3/88.6	89.5/86.1	91.0
	DeBERTaV3(0.3)	74.2	93.0	91.1	96.2	93.0	95.3	90.2	91.4	90.55	94.6/89.1	89.7/86.8	91.3
	DeBERTaV3(0.9)	74.4	93.0	90.9	96.0	93.0	95.3	89.5	91.7	90.48	94.2/88.5	89.9/86.9	91.2
Last 6	DeBERTaV3(0)	72.6	93.0	91.1	96.2	92.9	95.3	88.8	90.9	90.10	94.5/89.2	89.5/86.8	91.2
	DeBERTaV3(0.1)	72.9	93.0	91.1	96.2	92.9	95.3	88.8	90.9	90.14	94.5/88.9	89.5/86.7	91.3
	DeBERTaV3(0.3)	73.6	93.2	91.0	96.3	93.0	95.7	88.1	91.2	90.26	94.6/89.1	89.5/86.7	91.3
	DeBERTaV3(0.9)	74.2	93.1	90.9	96.0	93.0	95.4	88.5	90.9	90.25	94.7/89.0	89.5/86.8	91.2
Last 12	DeBERTaV3(0)	73.4	93.0	91.0	96.2	92.9	95.3	89.2	90.9	90.24	94.5/89.0	89.4/86.7	91.2
	DeBERTaV3(0.1)	73.9	93.0	91.0	96.2	92.9	95.5	89.9	91.1	90.44	94.4/88.9	89.5/86.9	91.2
	DeBERTaV3(0.3)	74.8	93.2	91.0	96.3	93.0	95.6	89.8	91.3	90.63	94.6/89.0	89.5/86.8	91.3
	DeBERTaV3(0.9)	74.2	93.1	90.9	96.0	93.0	95.3	89.3	90.9	90.34	94.7/89.0	89.4/86.7	91.2

suppress irrelevant features, weakening the influence of unnecessary information. A lower threshold Th has a weaker impact on inhibiting passing information, whereas a higher threshold inhibits most of the passing information.

Although the performance between LoRA and InA is similar, InA has the advantage of inhibiting unnecessary information by using an appropriate threshold. InA not only inherits the information compression ability of LoRA but also adds a mechanism to inhibit irrelevant information by applying the threshold. InA offers two key advantages over other adapters like LoRA and Adapter:

- (1) InA incorporates the rank of the adapter, r , to control redundant information flow through the bottleneck, effectively compressing the information.
- (2) InA uses a threshold to further limit the passing information, offering an additional control over the inhibition process. Thus, the passing information in InA is "incomplete" in the sense that task-irrelevant parts of the original information are discarded.

4.8.2 Should We Need Inhibition During Fine-Tuning? How Does It Work?

Redundant features obtained from pre-trained language models can reduce performance, especially when fine-tuning on small datasets. Therefore, we introduce a similar MLP architecture (inspired by gate multilayer perceptron (gMLP) [184]) combined with the proposed inhibition mechanism to address this challenge, which proves to be effective in reducing the impact of irrelevant knowledge.

We argue that InA is beneficial when fine-tuning pre-trained LMs on downstream NLU tasks. For instance, RoBERTa, pre-trained on over 160GB of text data with a larger mini-batch size and Byte-Pair Encoding [185], excels in handling large and diverse vocabularies

[160]. However, when applying InA to RoBERTa, it does not necessarily lead to better performance on tasks like RTE. We hypothesize that this is due to InA requiring less fine-tuning steps and tunable weights to scale the large, robust pretrained weights over smaller downstream tasks.

DeBERTa, which uses disentangled matrices for content and position vectors [163], has a stronger contextual connection among input word vectors. InA can inhibit redundant contextual information by scaling these disentangled matrices, effectively allowing DeBERTa to focus on the most relevant connections. In this way, InA aids DeBERTa by enabling it to concentrate on the most pertinent relationships in the data.

4.8.3 How to Choose the Inhibition Level Inh_p and Select a Good Rank r in Real Cases?

We investigate the effect of different inhibition levels (Inh_p) on fine-tuning tasks such as GLUE, SQuAD, and SWAG. From Table 5.5, we observe that an appropriate inhibition level (e.g., $Inh_p = 0.3$) improves text classification performance, while stronger inhibition (e.g., $Inh_p = 0.9$) benefits question-answering tasks.

In practice, when working with a smaller downstream dataset (e.g., RTE), we recommend initializing InA with 0% inhibition for the *Query* side. Alternatively, inserting InA into both the *Query* and *Key* sides with an inhibition level of 30% is also effective. Based on our experiments, we propose the following heuristic for selecting the appropriate inhibition threshold:

(1) Start with 0% inhibition, (2) If the performance improves over the baseline, choose an inhibition threshold between 10% and 30%, (3) If performance does not improve, increase the inhibition threshold (e.g., 90%).

For selecting the rank r in practical cases, we summarize the results of inserting InA into the last Transformer layers in Table 4.8. We find that inserting InA into a few layers does not significantly improve performance when fine-tuning DeBERTaV3 on downstream tasks. However, the best results are obtained by inserting InA into all layers or as many layers as memory allows.

4.8.4 Can InA Really Inhibit Irrelevant Knowledge? How Can It Do So?

To answer these questions, we focus on the performance of the inhibition vector $W_{inhibition}$ and its ability to suppress irrelevant knowledge.

When fine-tuning on the SQuAD task under five conditions: without InA, and with InA at $Inh_p = 0.0, 0.1, 0.3, 0.9$, we visualize the averaged attention score heatmap for the

last second layer (averaging all heads in the 23-th layer) in Figure 4.7. As the inhibition level increases from $Inh_p = 0.0$ to $Inh_p = 0.9$, the attention scores for "I" and "my" decrease, indicating that the influence of these irrelevant terms is gradually diminished. This suggests that InA effectively inhibits task-irrelevant knowledge during fine-tuning.

For the second question, we examine the influence of InA on attention scores across five downstream tasks: CoLA, RTE, MRPC, QNLI, and SWAG. From Table 5.1, we observe significant improvements on the CoLA task when InA is applied. For example, in Figure 4.6, the attention block primarily focuses on the relevant words such as ['They', 'him', 'to', 'by', 'making', 'him'], while irrelevant words like 'to' and 'by' are less emphasized after fine-tuning with InA. This demonstrates that InA reduces the impact of "noise" knowledge, making the classification process more efficient and accurate.

However, on the RTE task, InA does not outperform standard fine-tuning. We speculate that this is due to the small dataset size and the fact that InA eliminates knowledge that could potentially match the label in the entailment task. As shown in Figure 4.10, InA reduces the area of focus in the attention heatmap, eliminating unnecessary terms like "Slovenia", while retaining words that are highly relevant to the label, such as "3000" and "inhabitants". This demonstrates that InA enhances feature selection by inhibiting irrelevant information, although its effectiveness can depend on the dataset size.

4.9 Conclusion

We proposed an inhibition adaptation fine-tuning method—InA—which serves as a lightweight alternative that reduces the influence of irrelevant knowledge while maintaining high model performance. Specifically, InA retains the significant features of the model while eliminating both secondary task-relevant and task-irrelevant features, enabling quick task-switching properties when deployed in real-world services. There are several promising directions for future work:

(1) The mechanism behind InA fine-tuning, as discussed in this article, clarifies how InA inhibits task-irrelevant features while maintaining competitive performance on downstream tasks. However, on tasks like RTE, the retrieval of "irrelevant knowledge" and its alignment with the task label requires further investigation. Similarly, the application of InA to text generation tasks warrants additional exploration. Moreover, to recover the inhibited features, InA can be combined with other efficient adaptation methods (e.g., prefix-tuning, or other adaptations) that may re-enable the previously inhibited knowledge.

(2) Currently, the selection of weight matrices and inhibition levels for InA is primarily based on heuristics. A future direction would be to automate the selection of the inhibition

level during the fine-tuning process, enabling a more task-specific and dynamic adaptation for pre-trained language models.

(3) The activation function used in InA is another area for potential improvement. Investigating whether a more effective activation function could provide InA with a more suitable negative tail could lead to more refined inhibition behavior, which may be an important avenue for future research.

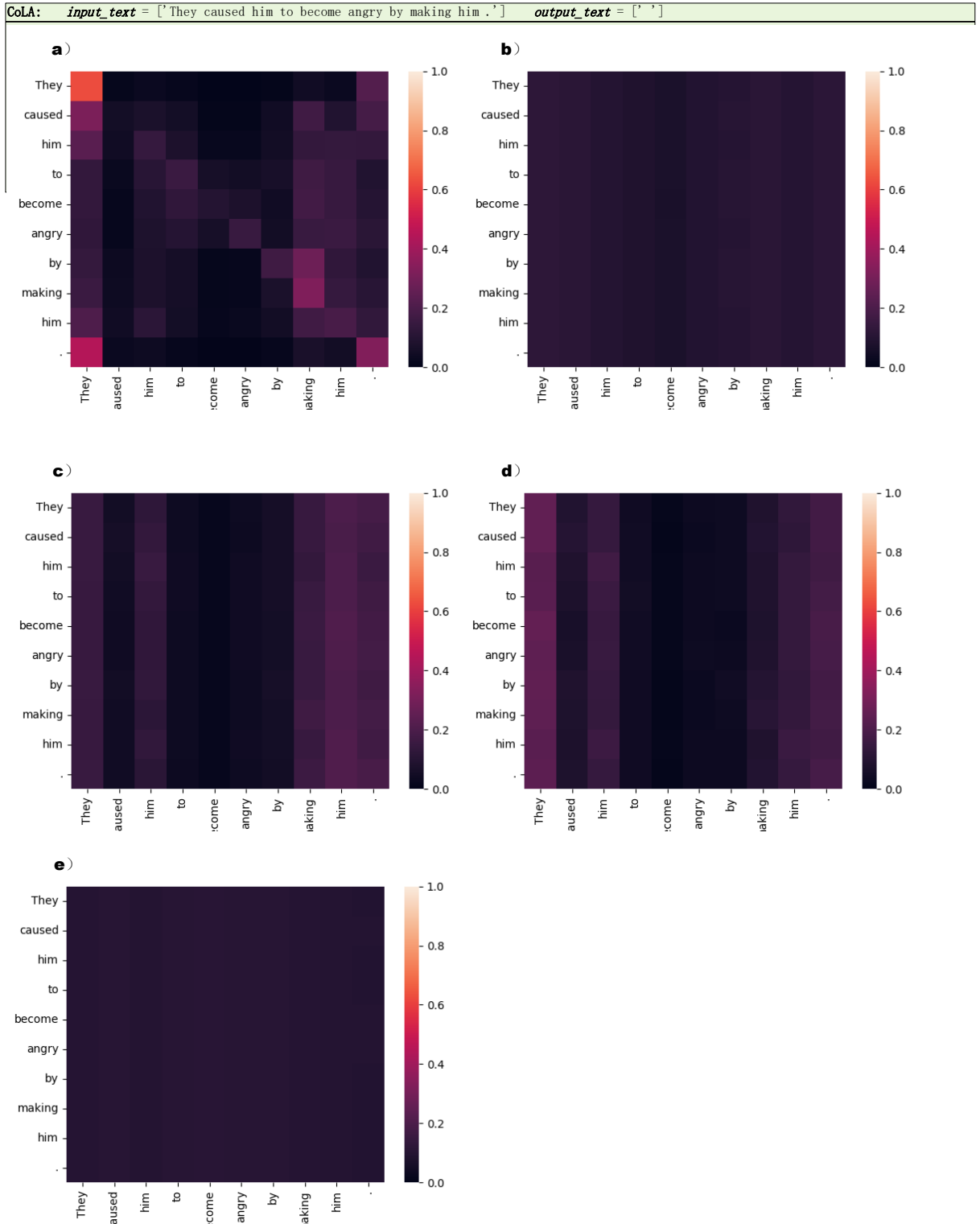


Figure 4.6: From left to right, fine-tuning *BERT* – *large* on CoLA with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), d) InA(0.9).

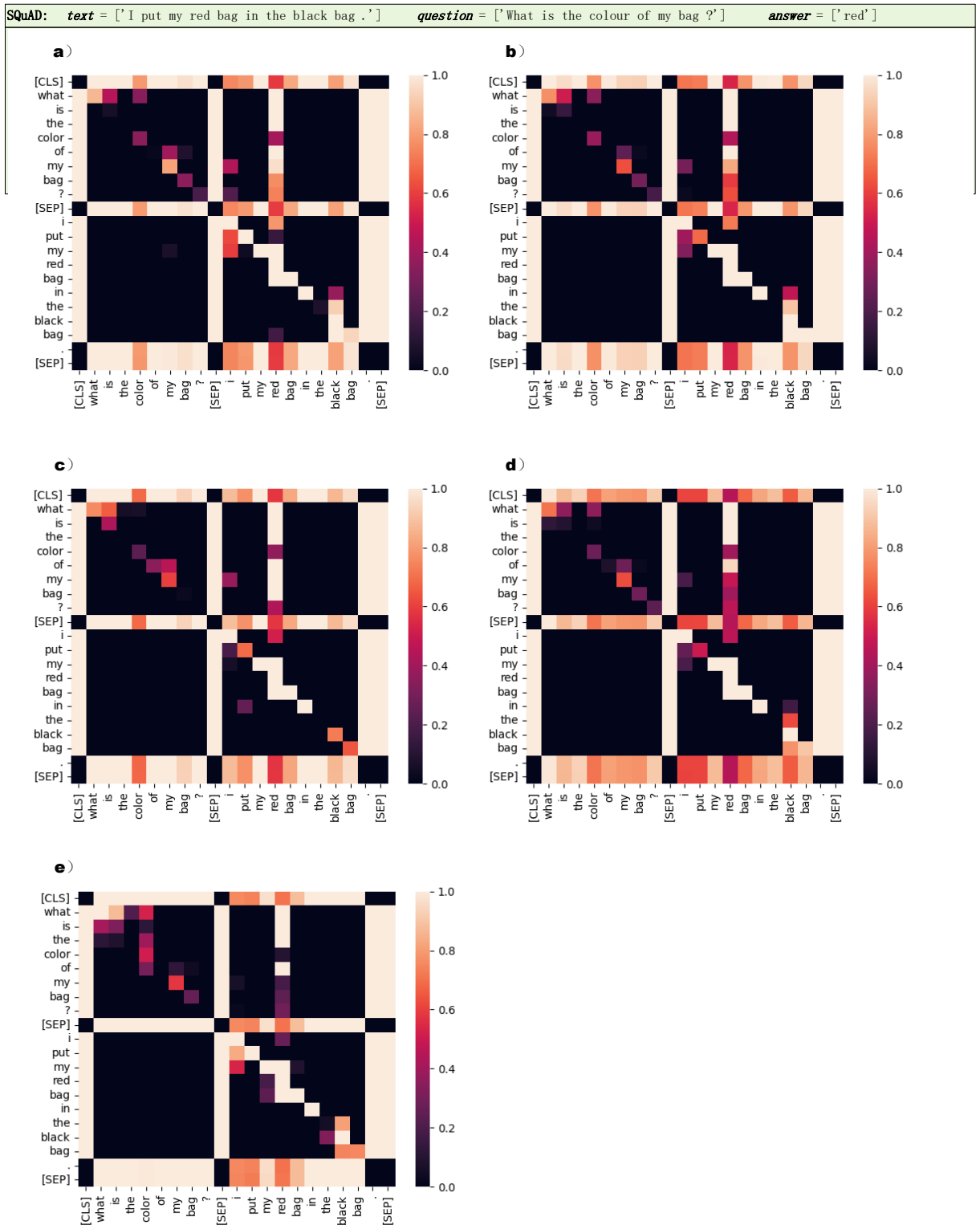


Figure 4.7: From left to right, fine-tuning *BERT* – *large* on SQuAD with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), e) InA(0.9).

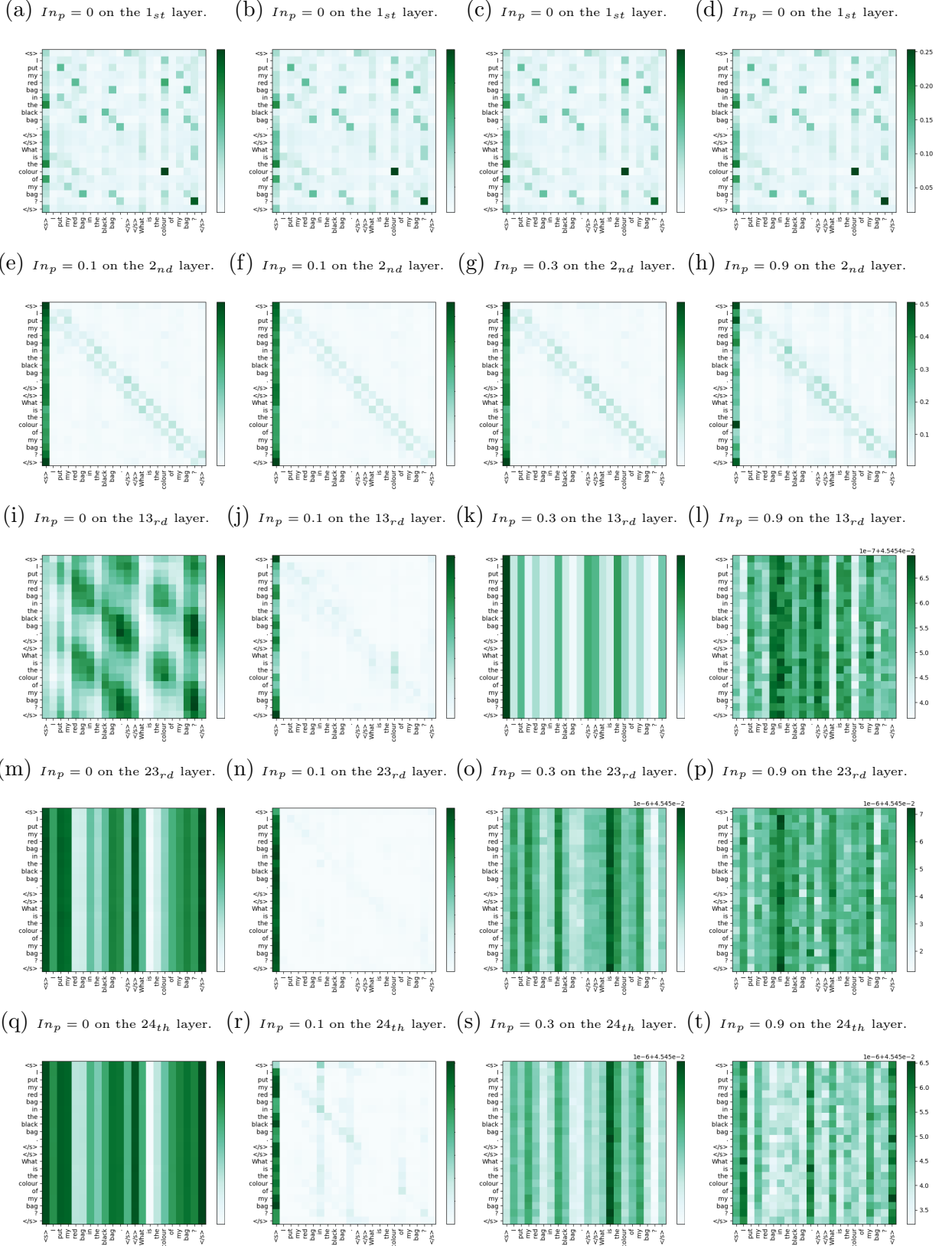


Figure 4.8: From left to right, fine-tuning *RoBERTa-large* on SQuAD-V2 with no-InA, InA($In_p = 0.0$), InA($In_p = 0.1$), InA($In_p = 0.3$), InA($In_p = 0.9$).

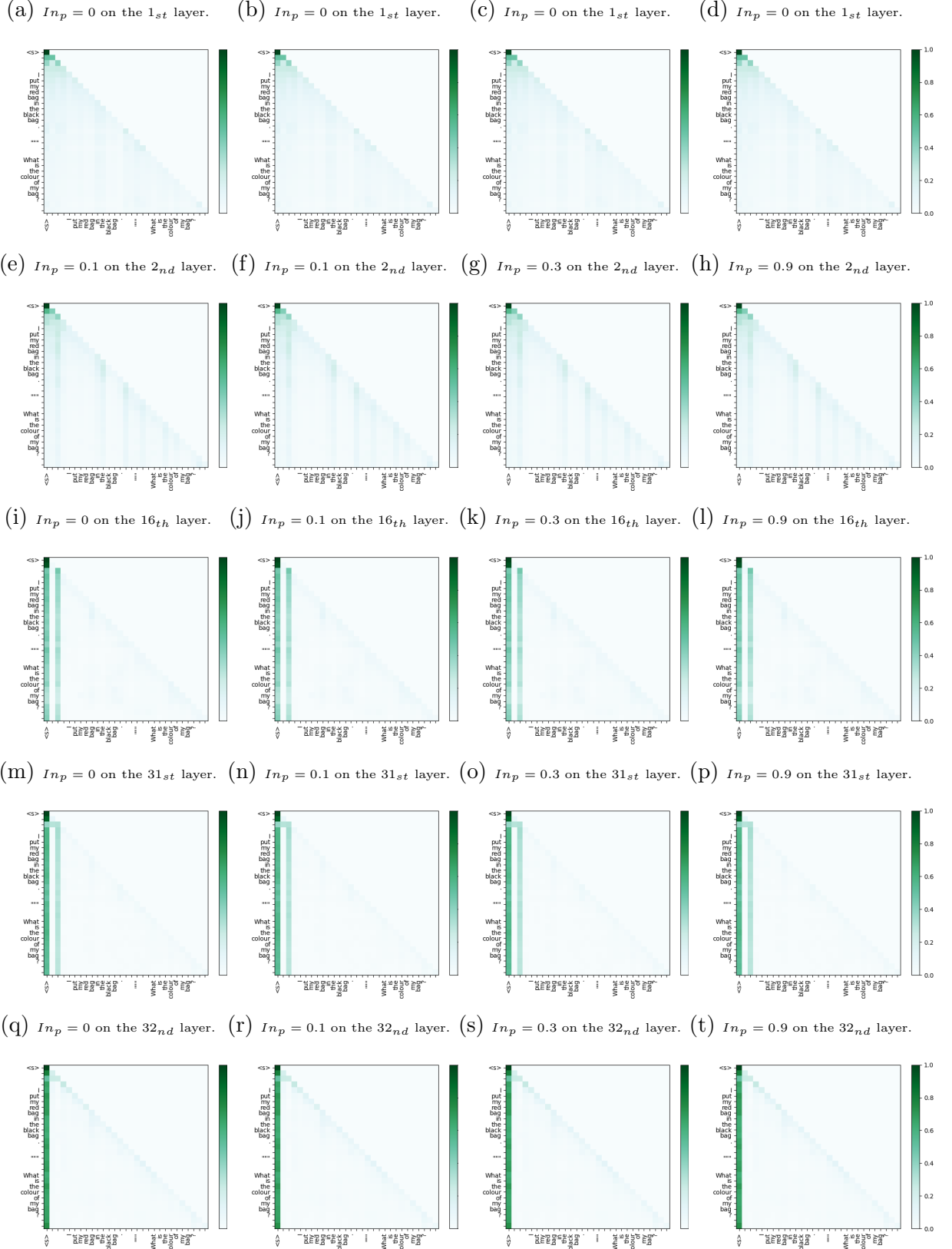


Figure 4.9: From left to right, fine-tuning *Llama2* on SQuAD-V2 with no-InA, InA($In_p = 0.0$), InA($In_p = 0.1$), InA($In_p = 0.3$), InA($In_p = 0.9$).

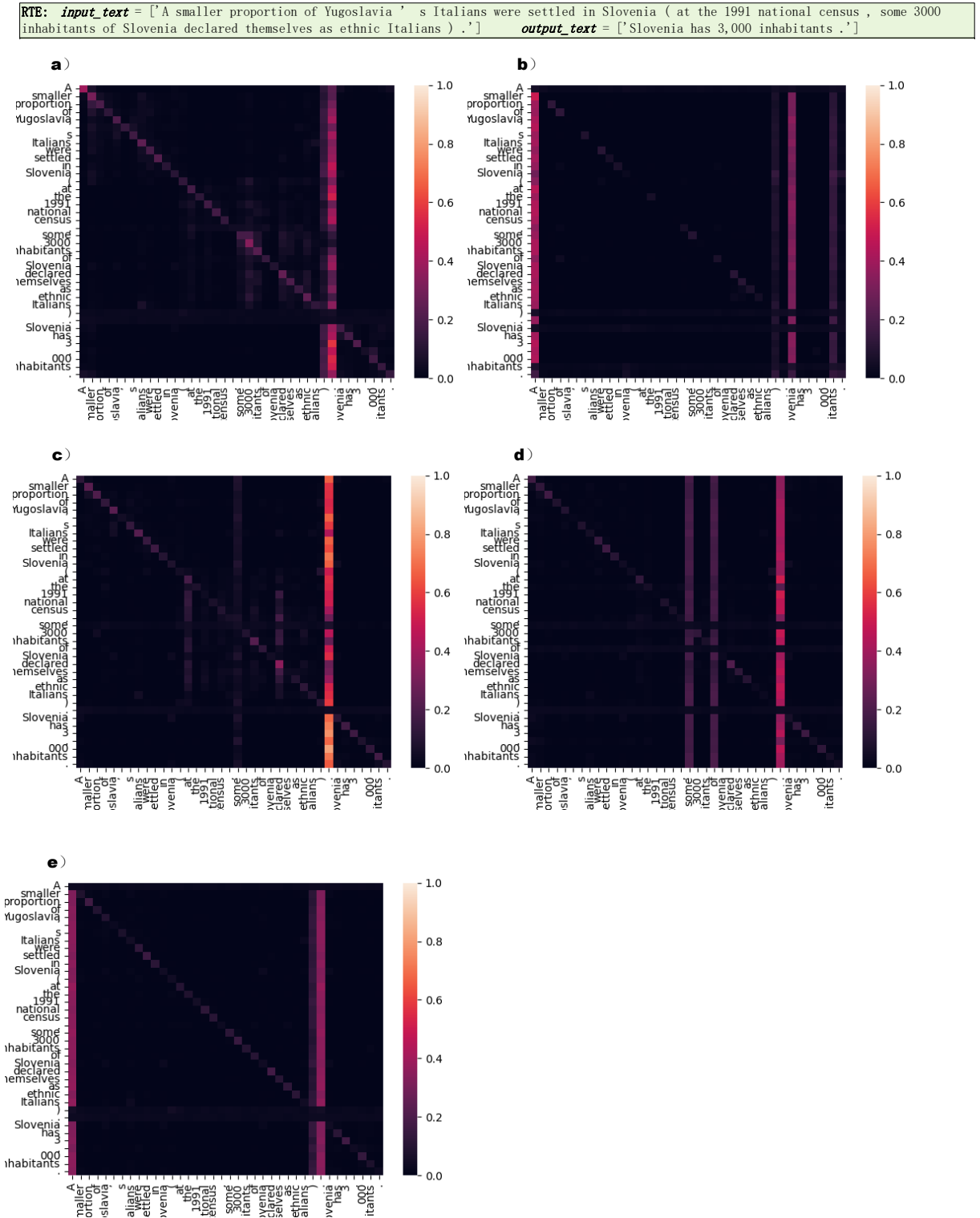


Figure 4.10: From left to right, fine-tuning *BERT* – *large* on RTE with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), d) InA(0.9).

MRPC: `input_text = ["We acted because we saw the existing evidence in a new light , through the prism of our experience on 11 September , " Rumsfeld said . '']` `output_text = ['Rather , the US acted because the administration saw " existing evidence in a new light , through the prism of our experience on September 11 " .']`

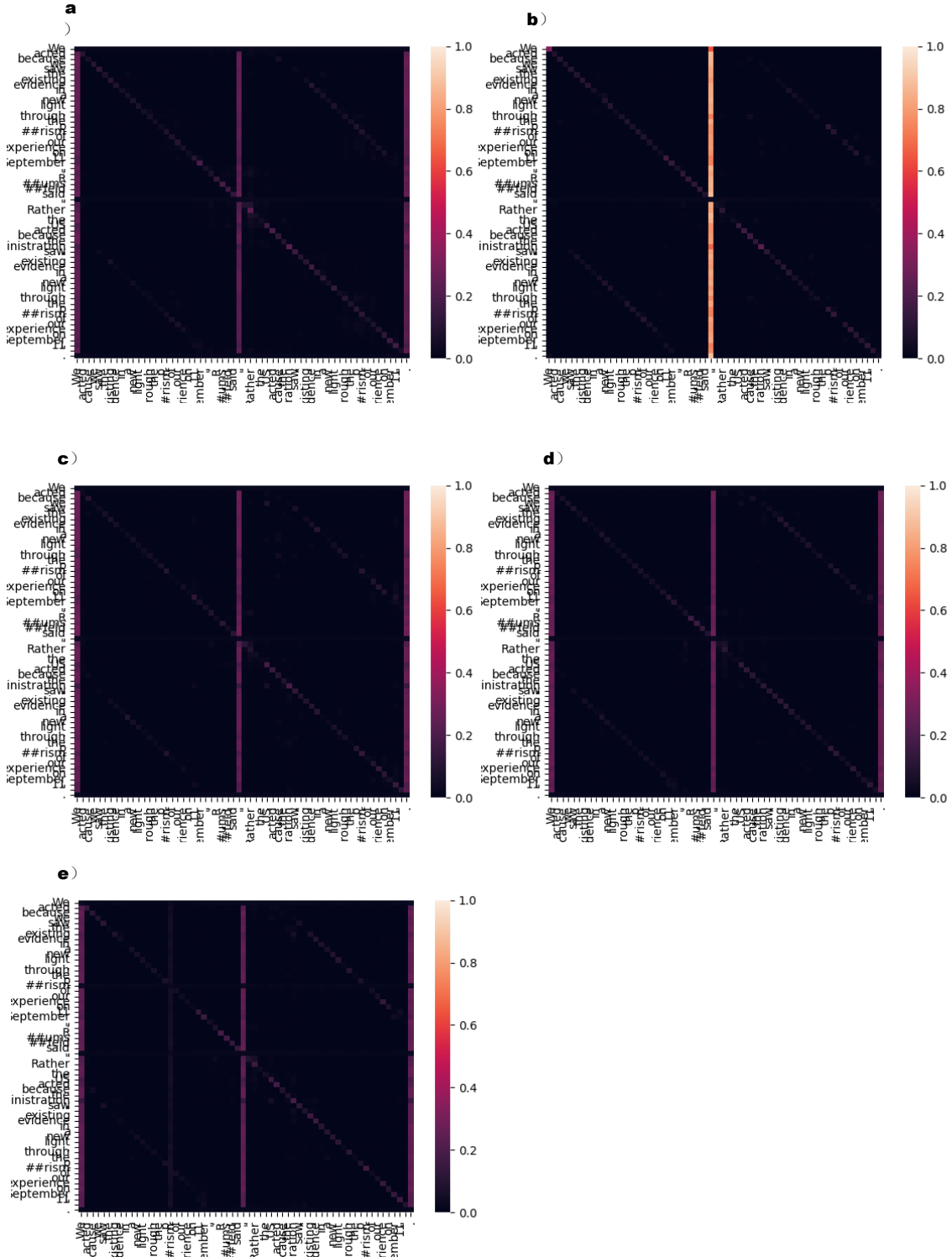


Figure 4.11: From left to right, fine-tuning *BERT* – *large* on MRPC with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), e) InA(0.9).

QNLI: `input_text = ['Where did Jebe die?']` `output_text = ['Genghis Khan recalled Subutai back to Mongolia soon afterwards, and Jebe died on the road back to Samarkand.']`

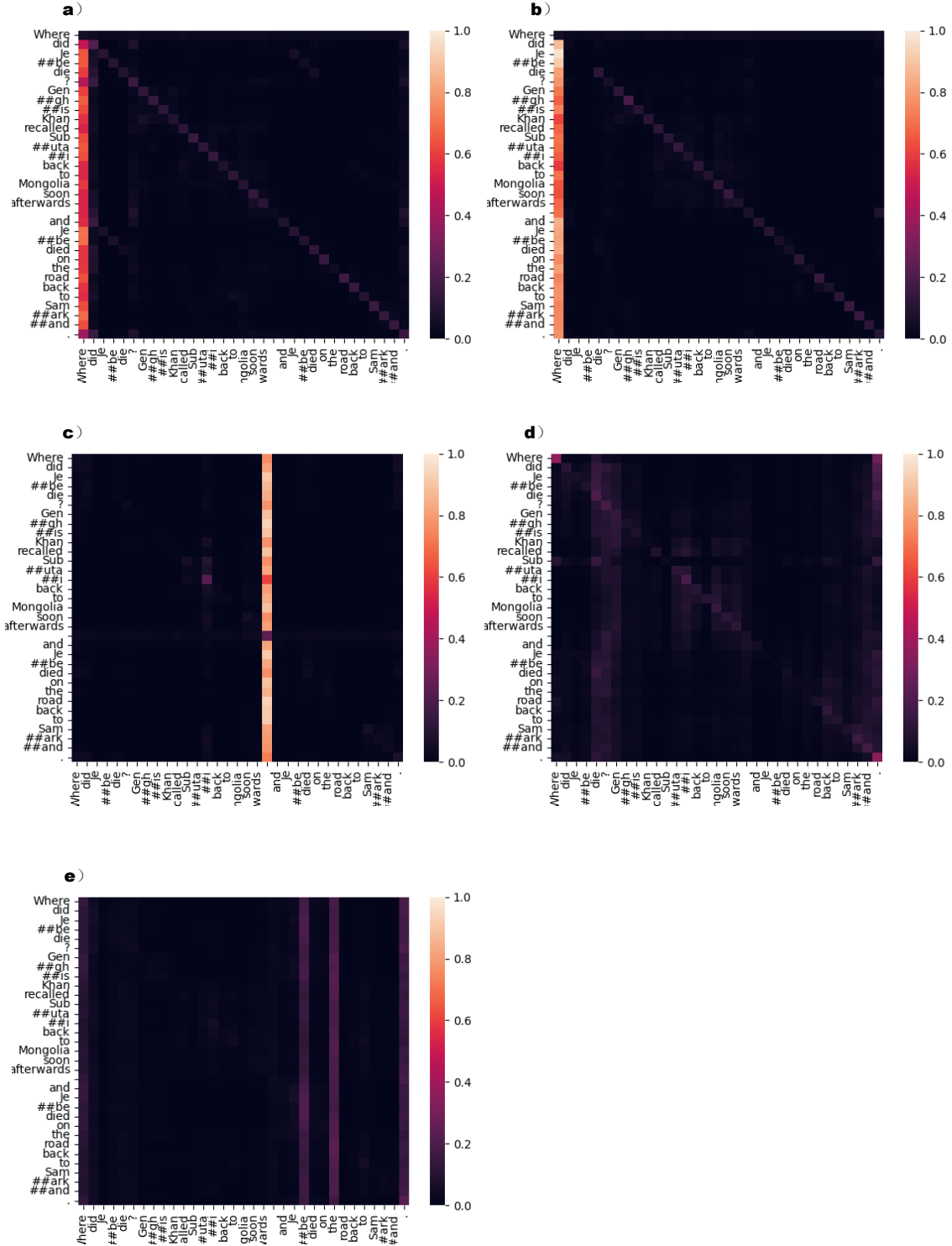


Figure 4.12: From left to right, fine-tuning *BERT* – *large* on QNLI with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), e) InA(0.9).

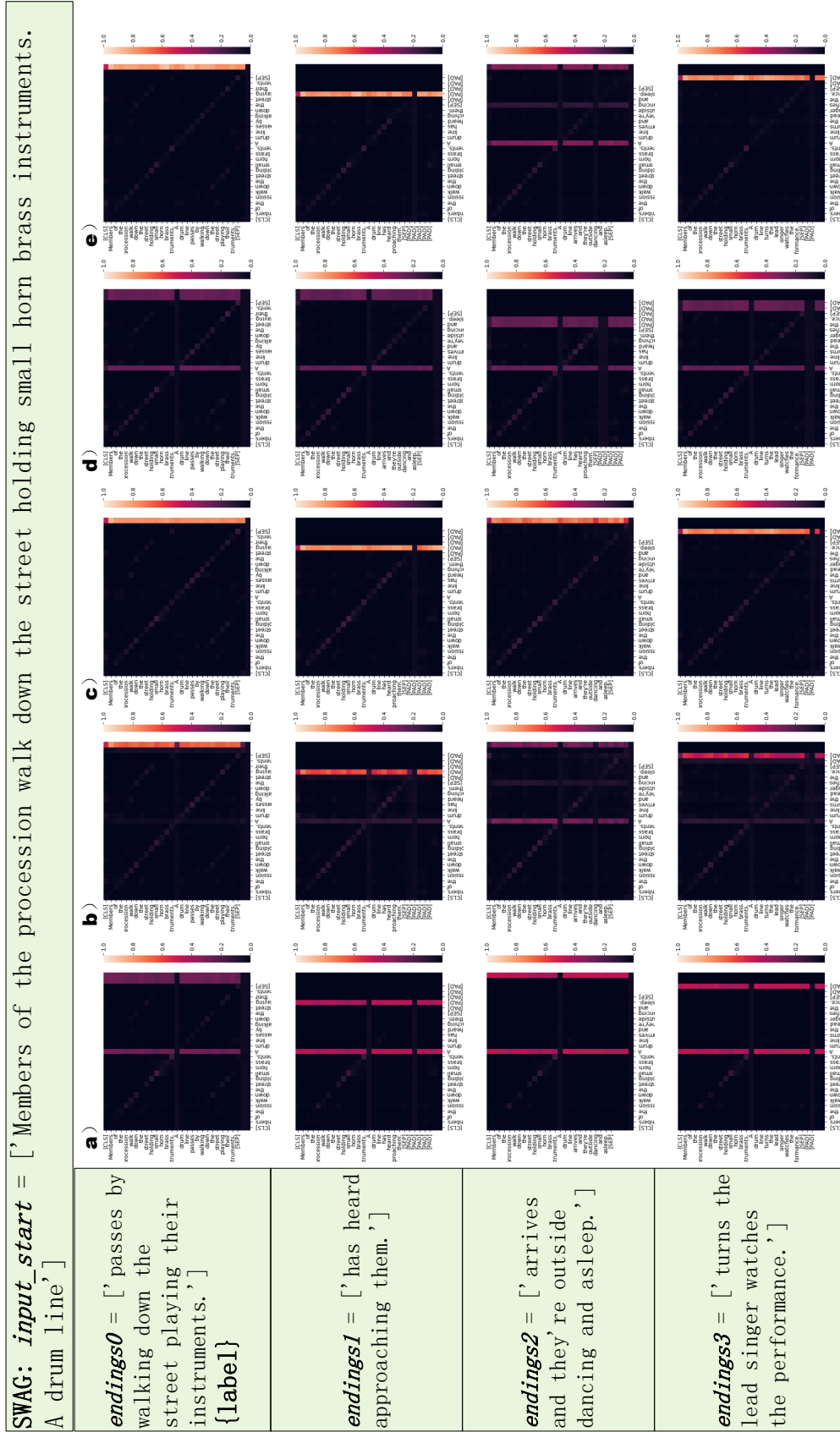


Figure 4.13: From left to right, fine-tuning *BERT* – *large* on SWAG with a) no-InA, b) InA(0.0), c) InA(0.1), d) InA(0.3), e) InA(0.9).

Chapter 5

Domain Specific Assistant Instruction For LLMs

LLMs have shown remarkable generalization capabilities across various tasks when provided with human-written instruction data. However, the limited quantity, diversity, and specialized nature of such instruction data raise concerns about the effectiveness of LLMs in specialized domains like psychotherapy. To address this, we propose two key solutions: first, we introduce Domain-Specific Assistant Instructions, grounded in the AlexanderStreet therapy dataset, and second, we employ an adaptation fine-tuning method combined with Retrieval-Augmented Generation (RAG) to enhance pre-trained LLMs. Through comprehensive quantitative evaluation of linguistic quality, incorporating both automatic and human assessments, we demonstrate that pre-trained LLMs fine-tuned with Psychotherapy Assistant Instructions significantly outperform SOTA LLMs response baselines. Our Assistant-Instruction framework offers a semi-supervised approach to align pre-trained LLMs with domain-specific instructions, thereby enriching these models with essential psychotherapy knowledge.

5.1 Introduction

LLMs have demonstrated impressive generalization capabilities, including in-context learning [51], chain-of-thought reasoning [52], and biomedical diagnostics [53]. Instruction tuning of LLMs has enabled them to follow natural language instructions and perform complex real-world tasks [54]. Two main methods have been developed for instruction-tuning LLMs: (1) fine-tuning the model on a diverse set of tasks using human-annotated prompts and feedback [55], and (2) supervised fine-tuning using public benchmarks and datasets augmented with manually or automatically generated instructions [56]. Additionally, RLHF has proven effective in improving LLMs in various domains, such as medicine

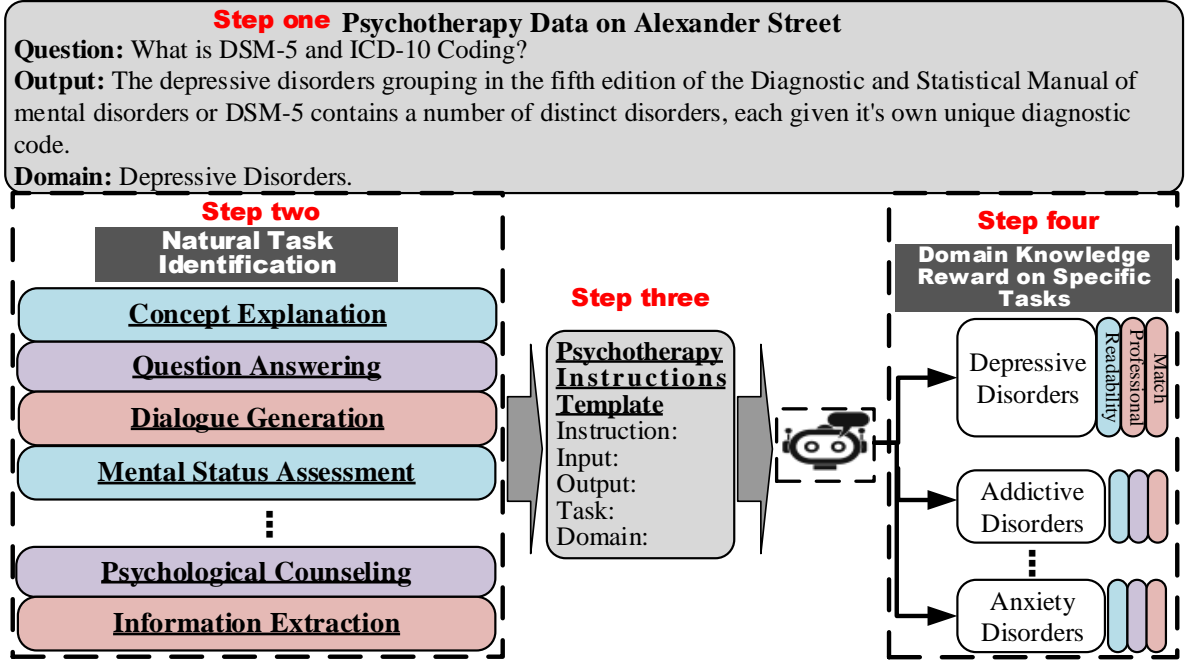


Figure 5.1: Schematic representation of Assistant-Instructional prompts in psychotherapy domains. Step one: Data reformatting; Step two: Task identification; Step three: Knowledge expansion; Step four: Evaluation.

[57], knowledge graphs [58], multimodal data fusion [186], and biomedical applications [59], although it is computationally expensive. Self-Instruct tuning [64], [65] and Guess-Instruction tuning have shown improved performance in aligning LLMs with human intent by learning from instruction-following data generated by advanced instruction-tuned teacher models (e.g., GPT-3, GPT-3.5, and GPT-4). These instruction-tuning approaches have been particularly effective in enhancing the zero- and few-shot generalization abilities of LLMs.

To advance the professional knowledge of LLMs in the psychotherapy domain, this paper presents the Psychotherapy Assistant-Instruction approach. Our method aims to (1) achieve generalization across various psychological consultation tasks and (2) incorporate specialized psychological knowledge into commonly used LLMs. Figure 5.1 provides an overview of our proposed approach, wherein a single model can perform multiple NLPs tasks within specific psychotherapy domains.

Evaluation of Domain-Specific Data: One of the most critical aspects of this work is the evaluation, particularly the role of domain-specific training data. In psychotherapy instruction-tuning, achieving human-level professional responses requires a novel approach. We propose using GPT-4 as an assistant for Assistant-Instruct tuning (a hybrid self-instruct tuning method) on psychotherapy consulting tasks (illustrated in Figure 5.2). Our method makes the following contributions: (a) it covers a broad spectrum

of psychological topics and integrates feedback from GPT-4-generated knowledge; **(b)** it incorporates psychotherapy knowledge from professional data, enabling the model to generate content closely aligned with GPT-4’s output; **(c)** it demonstrates the effectiveness of using assistant LLMs-revised instruction data to tune LLMs for psychotherapy tasks, providing practical insights for building general-purpose LLM-powered agents informed by assistant LLMs (e.g., GPT-4).

5.2 Problem Statement

The dataset we aim to generate consists of a collection of instructions $\{I_t\}$, where each instruction defines a specific domain t in natural language. Each domain t comprises $n_t \geq 1$ input-output instances $\{(X_{t,i}, Y_{t,i})\}_{i=1}^{n_t}$. We hypothesize that each domain t has unique characteristics (as illustrated in the left panel of Figure 1.4). The goal is for a model M to generate the correct output based on the domain-specific instruction and corresponding input: $M(I_t, X_{t,i}) = Y_{t,i}$, for $i \in \{1, \dots, n_t\}$.

The instruction is typically framed as: "Provide suggestions or comments on addressing and alleviating the following topic," and the instance input might be: "addictive disorders." However, there may be cases where the boundaries between instructions and input are not strictly defined. For instance, if the instruction is "Summarize the following description and explain the concept in the [***] domain. Add more common knowledge," and the input instance is "Addiction and Spiritual Crisis," the instruction domain may overlap with other domains. This overlap can make it challenging to construct instructions that are purely professional, as multi-domain knowledge could lead to instability during training, causing the model to incorporate irrelevant information.

To promote diversity and individuality in the data format, we allow instructions, input instances, and outputs to integrate additional knowledge from other models. Specifically, the output Y may be revised by GPT-4, such that $Y = Y + Y'$, where Y' represents the additional knowledge generated by GPT-4. As shown in the right panel of Figure 1.4, we face the challenge of making the data compatible with LLMs, where LLMs themselves are used to format instructions, inputs, and outputs, ensuring that the data remains both diverse and usable.

5.3 Related Work

5.3.1 Psychotherapy-based Conversational Systems

Chatbots are increasingly recognized for their ability to generate human-like social and emotional responses. However, their effectiveness as automated agents in domains like psychotherapy requires further investigation. Previous research has explored the potential and significance of integrating conversational AI into psychotherapy [187], [188]. Some studies have focused on the use of smart conversational agents to detect neuropsychiatric disorders [189], [190], employing deep neural networks to generate psychiatric-oriented responses. Other research [191] has examined the use of conversational agents in psycho-education and promoting self-adherence in mental health management. Furthermore, efforts have been made to fine-tune pre-trained language models on psychotherapy-specific datasets to improve their performance in the domain [192].

5.3.2 Instruction Data for Language Models

The annotation of large-scale instruction data presents several challenges for human annotators, especially in terms of **1) creativity**, required to generate novel domains, and **2) expertise**, needed to craft domain-specific solutions. To address these issues, several effective approaches have been proposed to generate, optimize, and reformat instructions.

Generate-Instruction: One approach for meta-training involves training the LM to generate task instructions from input instances and labels [193], [194]. During inference, the flipped learning method is used to train LMs by selecting the label option most likely to generate the corresponding task instruction. This method enables the generation of instructions from data in any format containing input instances and labels. However, a drawback of this method is that the generated instructions can deviate from the core task and fail to incorporate professional domain knowledge, such as that required for psychotherapy.

Self-Instruction: The Self-Instruction approach [56] offers a promising annotation-free method for aligning pre-trained LMs with instructions. It demonstrates the ability of LMs to generalize effectively to new tasks using GPT-3 and reformatting the generated instruction. This method involves concatenating the instruction with the instance input as a prompt and training the model to generate the output in a supervised manner. Multiple templates are used to encode both the instruction and the instance input to enhance the robustness of the model. Although Self-Instruction augments data without requiring annotations, it still lacks the ability to incorporate new, specialized knowledge, particularly in professional domains like psychotherapy.

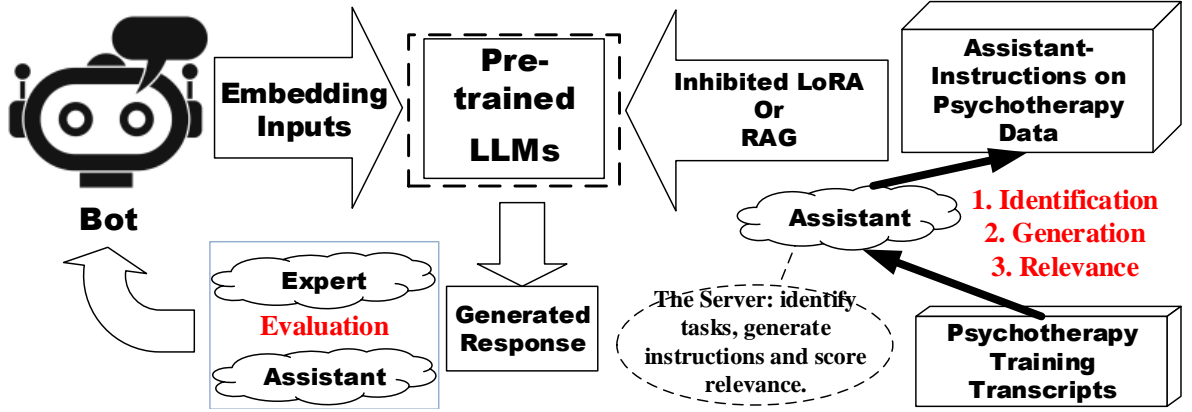


Figure 5.2: Schematic representation of model fine-tuning and the interaction between Chatbot and User.

Unnatural-Instruction: Unnatural-Instruction [60] is a large dataset of creative and diverse instructions that are generated with minimal human intervention. This dataset is created by prompting a language model with three seed examples of instructions and eliciting a fourth to create 64,000 unique examples. The dataset can be expanded further by prompting the model to rephrase each instruction, resulting in approximately 240,000 examples of instructions, inputs, and outputs. While this dataset is highly diverse, it does not effectively absorb new or professional domain knowledge, which limits its utility in specialized fields like psychotherapy.

In summary, while these approaches demonstrate promise in generating instructions, the main challenge remains the incorporation of new, domain-specific knowledge, particularly in fields requiring specialized expertise such as psychotherapy.

5.3.3 Parameter-Efficient Fine-Tuning of Pre-trained Language Models

Several [SOTA Parameter-Efficient Tuning Method \(PEFT\)](#) have been introduced, including Adapter [46], Prefix-Tuning [47], [LoRA](#) [48], [GLoRA](#) [170], and [InA](#) [7]. These methods focus on tuning only the added parameters while keeping the pre-trained language model frozen. They inject trainable low-rank matrices into transformer layers to approximate weight updates. Specifically, the methods update the Query, Key, and Value projection matrices (W_q , W_k , and W_v) in the multi-head attention sub-layer. Using a low-rank decomposition, the weight update is represented as:

$$H_o \leftarrow f(H \text{ concat } (P, W)) + s \cdot f(HW_{down} - Th)W_{up}, \quad (5.1)$$

where $H \in \mathbb{R}^{M \times d}$ represents the input hidden vectors, $H_o \in \mathbb{R}^{M \times d}$ is the output of

the self-attention layer, concat denotes the concatenation operation, f is the activation function, $s \geq 1$ is a tunable scalar hyperparameter, and Th is a threshold. The matrices $W_{down} \in \mathbb{R}^{d \times r}$ and $W_{up} \in \mathbb{R}^{r \times k}$ are used to update the model parameters.

These PEFT methods provide an efficient means to adapt pre-trained models to new tasks or domains, such as psychotherapy, without the need to fine-tune the entire model, thereby reducing the computational cost and memory requirements. The effectiveness of these methods in improving model performance for domain-specific applications, such as psychotherapy, is a topic of ongoing research.

5.4 Methodology

5.4.1 Data Collection

Alexander Street Press is a website known for its vast collection of video transcripts and recordings from therapy and counseling sessions, covering topics such as depression, abuse, trauma, and mental disorders. The video transcript dataset was specifically collected from the Counseling and Therapy channel on the website. We curated the dataset to include only English-language sessions recorded between 1980 and 2023, resulting in a set of 1,333 videos and accompanying transcripts. After filtering out short-length and non-informative videos, the final dataset comprises 1,179 video transcripts, containing a total of 188,421 dialogue turns. To ensure data quality, we performed a cleaning process to remove Unicode characters, pauses, and other unnecessary elements, resulting in a dataset with 3,141,520 words and a vocabulary size of 30,438.¹

On the Alexander Street Press website, most video transcripts and recordings consist of knowledge presentations and counseling talks. For knowledge presentations, there are no instruction questions or instance inputs, and the output is the content presented by the speaker. In the first step, we manually set instructions and instance inputs based on the discussed topics (e.g., Depressive disorders, Addiction, etc.). In the second step, we used the GPT-4 API to revise and generate instructions and instance inputs based on the contents.

5.4.2 Assistant on Annotation and Task Identification

To arrange psychotherapy data to correct tasks, such as (1) concept explanation and summarization, (2) question answering, (3) mental status assessment, (4) psychological counseling and (5) information extraction, (6) dialogue generation, (7) sentiment analysis,

¹<https://alexanderstreet.com/>

Table 5.1: Prompt used for identifying the type of tasks.

<p>Can the following task be regarded as a question answering task with finite output on [***] domain? Input: "JEFFREY MISHLOVE Yeah! Well we're running out of time... time. I supposed the point is that you've been successful in... in developing these devices and... and using them in the laboratory?" Output: "STEPHEN LABERGE Yes! That's right! Well, not just in the laboratory, but we developed devices that... that do have sensors built under the mass that could be used at home, so that... that of course was one of our major goals which used to have make lucid dreaming available in general to people so that they could make better lucid dream." Result: Yes</p>
<p>Can the following task be regarded as a dialogue generation task with finite output on [***] domain? Input: "CHRISTINA GROF Thank you... thank you. ,JEFFERY MISHLOVE It's a pleasure to be with you. ,CHRISTINA GROF It's a real pleasure to be here. " Output: "JEFFERY MISHLOVE When we think of addiction many people still don't feel that there might be a connection with... with the spiritual quest and in fact they almost seem as... as if they opposite ends of the spectrum, yet it was as... as you point out in the thirst for wholeness a phrase that comes from the great Swiss psychiatrist Carl Jung is it something that took place many decades ago, he... he recognized in a letter to the founder of Alcoholics Anonymous that... that there is a relationship between alcoholism and the spiritual quest." Result: Yes</p>
<p>Can the following task be regarded as a concept explanation task with finite output on [***] domain? Input: " ADDICTIVE DISORDERS " Output: "The human mind can be altered by any number of substances. These may be medications used to treat illness, infection, or even some of the mental disorders that we've looked at elsewhere in this series. They can be components of the food or beverages we consume that may change our cognition or behavior, or they can even be substances used to intentionally affect our behavior or perceptions of the world. However, any number of mind-altering substances can be used to the point that they cause clinically significant distress or functional impairment, leading to possible intoxication, abuse, addiction and withdrawal. This is the case for those who experience Addictive Disorders, also known as Substance-related and Addictive Disorders. These individuals may have their cognition, perceptions or physiology debilitated in reversible or irreversible ways. They may continue to use these substances, even when they experience clusters of negative effects. And they may become so dependent on a substance that even when they try to cease or reduce their use, they experience additional behavioral or cognitive consequences. Substance-related disorders are part of a larger grouping of disorders called substance-related and addictive disorders codified and defined in the fifth edition of the Diagnostic and Statistical Manual of Mental Disorders or DSM-5. These disorders share a common trait of activating the brain's reward system by a substance or addiction, and the individuals with these disorders continue their use and behavior, whether or not they recognize it as a problem. " Result: Yes</p>

(8) event ordering, we use an assistant LLM – GPT-4 to identify which task the human-constructed instruction should be. We directly prompt the LLM in a few-shot way to determine this, using 8 classification instructions from the seed tasks. The prompting template is shown in Table 5.1.

5.4.3 Assistant on Generation, and Evaluation

Our approach involves two main steps. Firstly, we optimize formulations that retain the content of the original instructions. We prompt a language model to reformulate the tasks in the core data for each generated task. In some instruction formulations, we embed the input into or add it behind the "INPUT" template – "We are talking about [***]." – to emphasize the topic. This manually constructed "INPUT" also captures the content discussed by members of the audience in Alexander Street Video, merging the discussed topic with the point of interest for the audience or visitors. Secondly, following [195], we use GPT-4 as an assistant to evaluate the retrieved passage's relevance. The prompting template is shown in Table 5.2.

Table 5.2: Prompt used for generation and evaluation.

<p>Prompt for Generation: "Make a more professional instruction and output based on given context of conversation in [***] domain. Remove people's names and UNKNOWN. Then, improve them all based on your knowledge. If you cannot do that, output nothing."</p> <p>Prompt for Evaluation: "Given an instruction and an output in [***] domain, rate whether the response appears to be a helpful and informative answer to the query, from 1 (lowest) - 5 (highest). The detailed criterion is as follows: 5: The response provides a complete, highly detailed, and informative response to the query, fully satisfying the information needs. 4: The response mostly fulfills the need in the query, while there can be some minor improvements such as discussing more detailed information, having better structure of the response, or improving coherence. 3: The response is acceptable, but some major additions or improvements are needed to satisfy users' needs. 2: The response still addresses the main request, but it is not complete or not relevant to the query. 1: The response is barely on-topic or completely irrelevant.."</p>
--

5.5 Experiments

5.5.1 Experiments Settings

We conducted an evaluation of the language models mentioned above for the task of response generation in the psychotherapy domain, specifically focusing on therapeutic counseling. The hyper-parameters used for querying the OpenAI API and fine-tuning LLMs in different experiments are respectively presented in Table 5.3 and Table 5.4. These hyper-parameters include batch size (bz), learning rate (lr), cut-off, inhibition percentile (Inh_P), hyper-parameters in InA (r , α , and $dropout$), temperature ($Temp.$) for controlling output randomness and diversity, top-p (Top_P) for limiting token selection, repetition penalty ($Penalty$), size of beam search algorithm ($Size_{Beam}$), and maximum output length ($Length_{Max}$). For generating the assistant instructions based on new psychotherapy data, we utilized the GPT-4 API as the Assistant-LLM. To fine-tune the generated instruction data effectively, we employed the inhibition adaption fine-tuning method on Llama2-7B and ChatGLM2-6B based on hyperparameters shown in Table 5.4. The fine-tuned LLMs were then evaluated by two psychologists on psychotherapy data. The fine-tuning process required two weeks for Llama2-7B and two days for ChatGLM2-6B when using four NVIDIA Tesla A100 GPUs with 40GB graphic memory cards. ²

We use a set of hyperparameters shown in Table 5.3 when querying GPT-4 API for different purposes. These hyperparameters are found to work well with the GPT-4 model.

Table 5.3: Hyper-parameters for querying OpenAI API in different experiments.

Experiments Settings	Self-Instructions Using GPT-4 API				
	$Temp.$	Top_P	$Penalty$	$Size_{Beam}$	$Length_{Max}$
Identifying Tasks	0	0	0	1	3
Generating Instances	0	0	1.5	1	512

Algorithm 1 describes the processing of psychotherapy data crawled from Alexander Street. We follow an iterative process to construct our own Assistant-Instruction set using GPT-4 and Self-Instruct [56].

²The code and data can be available at https://github.com/ChengKang520/psychotherapy-assistant_instruction

Table 5.4: Hyper-parameters for fine-tuning pre-trained LLMs in different experiments.

Experiments Settings	InA Fine-Tuning						
	<i>bz</i>	<i>lr</i>	<i>epochs</i>	<i>Inh_P</i>	<i>r</i>	<i>alpha</i>	<i>dropout</i>
Assistant-Instruction	128	0.001	40	0.3	32	16	0.05

Algorithm 1: Pseudo code for prompt engineering, GPT-4 call and hyper-parameters in data generation. The data flow is highlighted in blue.

Input: *prompt_input*, *prompt_no_input*.

```

1 prompt_input: (
2 "Make a more professional instruction, input and output based on the given context in [***]
  domain. \n\n"
3 "Remove people's names and UNKNOWN. Add more knowledge based on your knowledge. If
  you cannot do that, output nothing. \n\n"
4 "### Instruction: \n {instruction} \n\n ### Input: {input} \n\n ### Response:
  {response}"
5 ),
6 prompt_no_input: (
7 "Make a more professional instruction, input and output based on the given context in [***]
  domain. \n\n"
8 "Remove people's names and UNKNOWN. Add more knowledge based on your knowledge. If
  you cannot do that, output nothing. \n\n"
9 "### Instruction: \n {instruction} \n\n ### Response: {response}"
10 )
Output: output.
11 output = openai.ChatCompletion.create (
12 model="chatgpt-turbo",
13 messages [ "role": "user", "content": prompt ],
14 temperature = 1.0,
15 top.p=1.0, # nucleus sampling over entire vocabulary
16 max_tokens=512, # the max number of generated tokens
17 )

```

5.5.2 Models

We conducted an evaluation of the language models mentioned above for the task of response generation in the psychotherapy domain, specifically focusing on therapeutic counseling. For generating the assistant instructions based on new psychotherapy data, we utilized the GPT-4 API as the Assistant-LLM. To fine-tune the generated instruction data effectively, we employed the inhibition adaption fine-tuning method [3] and self-RAG [195] on Llama2-7B [62], as well as ChatGLM2-6B [196]. The fine-tuned LLMs were then evaluated by two psychologists on psychotherapy data.

5.5.3 Metrics

As shown in Table 5.7, our fine-tuned LLMs are evaluated under two main methods: Automatic evaluation and Human evaluation. For the automatic evaluation, we submit 200 generated answers and use Rouge-L [197] and Fluency [198] to compare the generated output with a reference text and calculate a score for the similarity evaluation. For the

Table 5.5: The manually constructed Instruction and GPT-4 revised Assistant-Instruction on the Depressive Disorder domain.

Instruction: What suggestions or comments you can provide to address or alleviate the following topics? Input: We are talking about [Depressive Disorders]. Output: A major depressive episode has a number of characteristic features, which take place most of the day, nearly every day, where relevant. The two main characteristics are a depressed mood, and a noticeably decreased interest in most activities. Other characteristics include trouble sleeping, diminished hunger or pronounced weight loss, trouble concentrating or making decisions, fatigue or loss of energy, recurring suicidal thoughts with or without intent to act on these thoughts, feelings of worthlessness or excessive or inappropriate guilt, psychomotor agitation or retardation, and feelings of low self-worth or pronounced guilt. These symptoms must cause marked distress in the patient's work or social life. For all of these episodes the clinician should ensure that the symptoms are not better explained by another medical condition or by the affects of a substance.	Instruction: Kindly provide professional suggestions or comments on effectively addressing and alleviating [Depressive Disorders]. Input: We are discussing [Depressive Disorders]. Output: A major depressive episode is characterized by a range of distinct features that persist for most of the day, nearly every day. The primary indicators include a persistent low mood and a significant decrease in interest or pleasure in most activities. Other symptoms may involve sleep disturbances, changes in appetite or weight, difficulty concentrating or making decisions, fatigue, recurring thoughts of suicide with or without intent to act on them, feelings of worthlessness or excessive guilt, psychomotor agitation or retardation, and feelings of low self-worth. These symptoms should cause marked distress in the individual's work, social life, or daily functioning. When addressing Depressive Disorders, clinicians should ensure that the symptoms are not better explained by another medical condition or the effects of substance use. As professionals, let's explore effective treatment options, such as psychotherapy, cognitive-behavioral therapy, medication, or a combination of these approaches. Encouraging patients to engage in regular physical activity, maintain a healthy diet, and establish a supportive social network can also be beneficial. Furthermore, it is crucial to foster a compassionate and understanding environment to help individuals with Depressive Disorders seek appropriate help and support.
Perplexity: 6.71	Perplexity: 2.15

human evaluation, we also submit 60 generated outputs (20 examples for each tasks, as we have 3 main tasks) without model and domain information to two professional psychologists, and based on their professional psychotherapy knowledge, they will measure the quality of the generated content according to the questions and answers. There are six grades that they will use to mark the quality of generated content: Extremely Bad (1), Bad (2), Neutral (3), Acceptable (4), Good (5) and Very Good (6). They should consider the readability, professional and match score (that measures the match with the domain knowledge).

5.5.4 Analytic Experiments

Performance on Revision

Due to numerous vocabulary and grammar errors in the original psychotherapy data, manual processing and revision of these errors were not feasible. In Table 5.1, we presented the one command that requires generated instructions from GPT-4. The right panel of Table 5.5 contains additional information contributed by GPT-4, such as common knowledge about depressive disorders, including psychotherapy, cognitive-behavioral therapy, medication, and a combination of treatment approaches. These recommendations from GPT-4 are considered common-sense by psychologists, but the original psychotherapy data lacked

Table 5.6: Based on Llama2-7B, we illustrate the performance of Zero-Shot, inhibited LoRA Tuned and RAG methods on Psychotherapy data.

Instruction: Please provide professional suggestions or comments to effectively address and alleviate [Autism].
Input: We are discussing [Autism].
Output from Zero-Shot: Autism is a developmental disorder that can cause significant social, communication and behavioral challenges. There is no known single cause for autism, but increased awareness and funding support research into possible causes and better ways to treat the disorder.
Output from Inhibited LoRA: Autism is a developmental disorder that can cause significant social, communication and behavioral challenges. It is a spectrum disorder, which means that symptoms and severity vary widely among individuals.
Output from Inhibited LoRA + RAG: Autism is a developmental disorder that can cause significant social, communication and behavioral challenges. It is a spectrum disorder, which means that symptoms can range from mild to severe. Some people with autism may also have other conditions, such as attention deficit hyperactivity disorder (ADHD), anxiety disorders, or epilepsy. Applied Behavior Analysis (ABA) a widely recognized and evidence-based therapy for Autism. ABA techniques focus on promoting positive behaviors and reducing challenging behaviors. Speech and language therapy can help improve communication skills, such as speech articulation, expressive language, and social communication.

sufficient common knowledge (because the original psychotherapy data merely has professional knowledge). The instructions and instance inputs were also refined by GPT-4 based on the given command. Perplexity is a metric that gauges the language model’s ability to predict a sequence of words. The perplexity of the right panel is lower than that of the left panel. This revision process makes the content more LLM-friendly, transforming ”professional knowledge format” into a more accessible ”common knowledge format” (or open-domain knowledge format).

Generation on Psychotherapy Domain

When comparing to the content generated by pre-trained ChatGLM2-6B, we observed that the RAG method contains additional professional knowledge (can be seen from Table 5.6). While this extra knowledge provides more professional information, pre-trained LLMs could not offer further professional insights into autism. When comparing the generated content of ChatGLM2-6B fine-tuned on inhibited LoRA to that of ChatGLM2-6B, we found that the former provides more refinedly revised information (However, this difference is not obvious). For example, it offers insights on professional explanation, such as ” *It is a spectrum disorder, which means that symptoms and severity vary widely among individuals.*”.

Evaluation

We present a performance summary of different instruction-tuning methods applied to two pre-trained LLMs in Table 5.7. While the Rouge-L and Fluency evaluation results show improvement with the use of Assistant-Instruction. To validate the performance, we

Table 5.7: For evaluating the performance of LLM on psychotherapy domain, two methods - inhibited LoRA and RAG - were used on two pre-trained LLM have been tuned on Assistant-Instruction using .

Inhibited LoRA Finetuning (without / with Assistant-Instruction)					
Pretrained LLM	Automatic		Human Evaluation		
	Rouge-L \uparrow	Fluency \downarrow	Read	Prof	Match
ChatGLM2-7B	24.3/27.1	49.4/48.7	4.8/4.9	2.9/3.3	2.1/2.5
Llama2-7B	15.1/16.9	20.9/20.5	5.0/5.2	3.0/3.2	1.9/2.3
Retravel Augmented Generation (without / with Assistant-Instruction)					
Pretrained LLM	Automatic		Human Evaluation		
	Rouge-L \uparrow	Fluency \downarrow	Read	Prof	Match
ChatGLM2-7B	25.1/32.8	56.4/46.7	4.6/5.3	3.9/4.2	2.9/3.3
Llama2-7B	15.4/22.4	30.3/20.7	4.8/5.2	3.7/4.1	3.0/3.4

use a selected portion of psychotherapy data as a validation set. Through content revising and leveraging additional common knowledge from GPT-4, both of these two LLMs show significant enhancement in matching the revised answers. Pre-trained LLMs can provide clients with comments to address psychological problems, but the quality of generated content may not always be fully accepted by psychologists. From Table 5.7, we observe that psychologists tend to prefer models that have been fine-tuned on psychotherapy data. As most LLMs lack specialization in a specific domain, they often require more domain-specific knowledge to improve their performance in professional domains. Because LLMs have been pre-trained on a vast corpus, giving them an inherent advantage in readability, and the size of tokens used does not seem to affect their performance significantly. Regarding the professionalism of the generated content, the psychologists gave higher scores to models that had been fine-tuned on psychotherapy instruction data compared to the corresponding original LLMs.

Human Evaluation Agreement

To assess the reliability of our human evaluation, we conducted an inner-rater agreement analysis [56] between our two evaluators. We used Cohen’s κ to measure inter-rater agreement for categorical items. The 6-level rating scale (ranging from 1 to 6) was treated as a categorical variable for each aspect under consideration. The resulting κ value was 0.63, indicating a moderate level of agreement according to common practice. Furthermore, we computed the Spearman correlation coefficient ρ between the ratings of our two evaluators, treating the ratings as ordinal variables (ranging from 1 to 6). The obtained coefficient was $\rho = 0.81$, demonstrating a high correlation between the two evaluators. These results indicate a reasonably reliable human evaluation process for our study.

5.6 Conclusion

We propose a novel method called ASSISTANT-INSTRUCT for fine-tuning or retrieving information from LLMs to improve their instruction-following ability. This method combines both common knowledge and psychotherapy professional knowledge to generate instruction data with the help of experts. It retains the general knowledge already present in pre-trained LLMs and incorporates psychotherapy-specific knowledge from expert-presented instructions. To enhance fine-tuning, as well as retrieval knowledge, we format the psychotherapy data, such as presentations, talks, and conversations, to make it more compatible with LLMs. Human evaluation of this method demonstrates significant improvement compared to existing instruction methods. ASSISTANT-INSTRUCT can serve as an initial step to align pre-trained LLMs with LLM-revised instructions, and further research can build upon this method to enhance instruction-following models.

Chapter 6

LLM-ABBA For Digital Health

The success of [LLMs](#) in the time series domain has been demonstrated through various benchmarks. By utilizing symbolic time series representations, it is possible to effectively bridge the gap between [LLMs](#) and time series data. However, the remaining challenge lies in exploiting the semantic information embedded in time series through symbols or existing tokens of [LLMs](#), while simultaneously aligning the [LLMs](#) embedding space with the domain-specific information inherent in the time series data. The [Symbolic Time Series Approximation \(STSA\)](#) method, known as [ABBA](#), has shown exceptional efficacy in preserving key time series features by modeling time series patterns in terms of amplitude and period, while leveraging existing tokens of [LLMs](#).

In this paper, we introduce a method called [LLM-ABBA](#), which integrates [ABBA](#) with large language models for various downstream time series tasks. By symbolizing time series data, [LLM-ABBA](#) outperforms recent [SOTA](#) methods in some UCR data and three medical time series classification tasks. Additionally, a fixed-polygonal chain technique is introduced within [ABBA](#) to prevent significant drift during forecasting tasks by mitigating the effects of cumulative errors caused by misused symbols during the transition from symbolic to numerical values. In time series regression tasks, [LLM-ABBA](#) sets a new [SOTA](#) on [Time Series Extrinsic Regression \(TSER\)](#) benchmarks. Furthermore, [LLM-ABBA](#) demonstrates competitive prediction performance compared to the latest [SOTA](#) results in time series prediction. We believe this framework can be easily extended to other time series domains as well.

6.1 Introduction

Time series are fundamental mathematical objects with applications across a wide range of disciplines, such as classification [199], regression [200], and prediction [201]. Recently, the potential of [LLMs](#) in time series applications has been increasingly recognized. A

recent review [202] identifies three main LLM-based approaches for learning complex semantic and knowledge representations from time series to perform various tasks. The first approach involves patching and tokenizing numerical signals and related text data, followed by fine-tuning on time series tasks [67], [203], [204]; the second approach preprocesses time series data to fit LLM input spaces by adding a customized tokenizer [205]; and the third approach builds foundational models from scratch, aiming to create large, scalable models that are both generic and domain-specific [206], [207].

Each of these techniques has its own limitations. Patching and tokenizing time series segments enables the mapping between time series data and the latent embedding of LLMs, but it requires generating numerical values digit by digit, which slows down the generation speed [204]. Additionally, while adding a customized tokenizer allows LLMs to handle the positions of time series patterns and reproduce the internal logic of time series signals [208], tokenizers not designed for numerical values separate continuous values and ignore the temporal relationships in time series. This method therefore necessitates converting tokens into flexible continuous values [209], which introduces the risk of semantic loss during the transition from the time series feature space to the latent embedding space of LLMs. Building foundational time series models from scratch can address some of these issues, but it remains challenging due to the high development costs and expensive training requirements [202].

By aligning time series with natural language, large language models and specialized time series models form a new paradigm where LLMs are prompted with both time series data and text-based instructions [202]. In this paradigm, time series and textual information provide essential context, LLMs contribute internal knowledge and reasoning capabilities, and

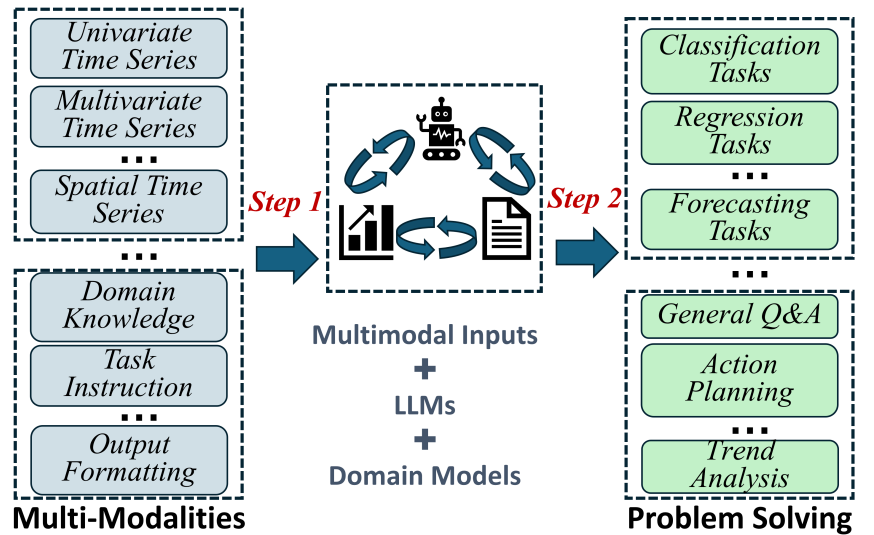


Figure 6.1: The integration of time series and LLM demonstrates potential in solving complex real-world problems.

time series models offer reliable pattern recognition. This novel integration is depicted in Figure 6.1, where the successful combination of these components demonstrates the

potential for a unified, general-purpose system for next-generation time series analysis. The challenge, however, lies in developing a tool that can transform the internal patterns of time series into a form that LLMs can recognize (*Step 1* in Figure 6.1). Moreover, this tool must also be capable of transforming the generated content back into the time series domain to support time series analysis (*Step 2* in Figure 6.1).

STSA is a method that converts time series data into symbols, establishing a bridge between strings and numerical time series. This enables the Chain-Of-Pattern (COP) of strings to carry more information than raw data. By symbolizing time series, one can model them as native languages, encoding them as a sequence of strings and applying efficient text analysis techniques, such as converting time series forecasting into next-token prediction in text. STSA aligns time series features with symbols both implicitly and explicitly, allowing for the manipulation of natural language processing techniques on time series data. Ideally, this eliminates the need to (1) patch and tokenize time series segments, (2) add an extra customized tokenizer set, or (3) build foundational time series models from scratch. Symbolic representations of time series can reveal the linguistic logic hidden within the signals, providing LLMs with the ability to understand temporal patterns. Inspired by this idea, the goal is to develop a method that can efficiently transform numerical time series into symbols and fine-tune LLMs for time series analysis tasks (e.g., classification, regression, and prediction).

However, integrating STSA methods with LLMs remains a challenge. Applying LLMs to symbolic time series representations introduces difficulties. First, we must address the issue of symbolic consistency in STSA methods, as symbols representing the same concept in different time series should remain consistent. It is also unclear whether LLMs will learn consistent knowledge from transformed symbols that encode time series pattern logic. Additionally, while LLMs can generate text based on given information, it is uncertain whether they can generate symbolic series and reconstruct time series pattern logic via STSA methods. These challenges lead us to ABBA [71] (including its accelerated variant Fast Adaptive Brownian Bridge-based symbolic Aggregation (fABBA) [72]), the most recent STSA method, which offers a competitive advantage in capturing the shape of time series data compared to other STSA methods. Unlike other methods, ABBA allows users to define custom strings for symbolization and provides open-source software with user-friendly APIs¹. Each ABBA symbol is linked to a unique real-valued cluster center, enabling a natural word embedding for symbols, akin to a native language. The effectiveness of STSA methods can be evaluated by visualizing their symbolic reconstruction. A comparison of reconstruction using Symbolic Aggregate approXimation (SAX) [210] and fABBA [72] is shown in Figure 6.2. It is clear that SAX fails to capture the

¹<https://github.com/nla-group/fABBA>

time series trend in both figures, and the peak information in figure (b) is missing in the SAX reconstruction. In contrast, fABBA better captures the essential information of time series patterns.

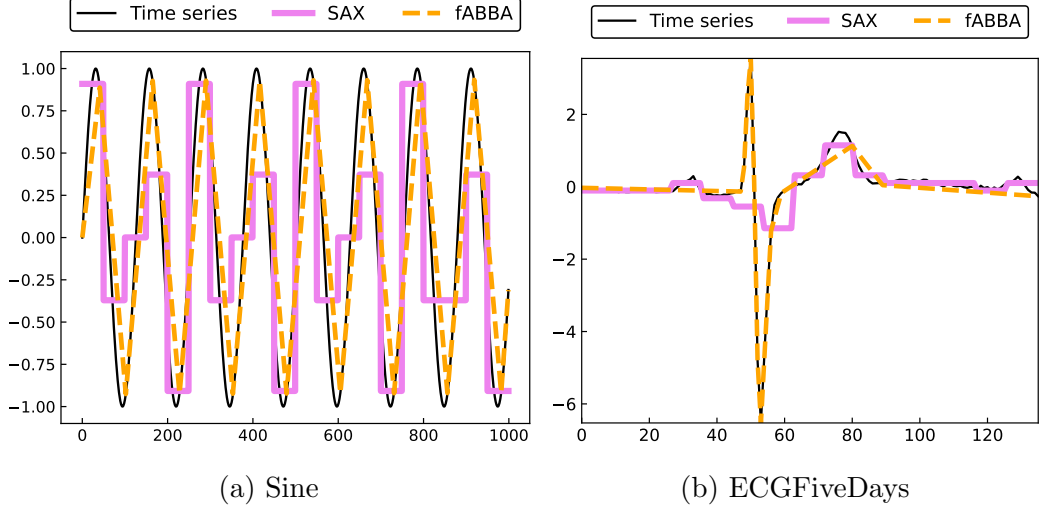


Figure 6.2: Plot (a) shows a sine function with 1,000 points, and (b) shows the ECG-FiveDays time series from the UCR Archive. We first perform fABBA with $\text{tol}=0.1$ and $\alpha=0.1$ and perform SAX with approximately the same length of symbolic representation and the number of distinct symbols. In plot (a), fABBA generates symbols “aB-bCbCbCbCbCbCA” while SAX generates symbols “aACBbaACBbaACBbaAABb”; in figure (b), fABBA generates symbols “fAcaDECeBdbF” while SAX generates symbols “AAAAAABbCcDaaaAaa”.

In this paper, we propose LLM-ABBA, a method that enables LLMs to understand time series data by using ABBA to transform numerical time series signals into symbolic series. Specifically, LLM-ABBA first converts time series signals into compressed representations by adaptively compressing the numerical inputs. These compressed representations are then digitized with predefined symbols or pretrained tokens. LLM-ABBA provides LLMs with a series of symbols (or pretrained tokens) that they can recognize, and these symbols encapsulate the COP of the time series signals. For classification tasks, the goal is to identify the symbolic series, while for forecasting or regression tasks, an additional step is taken to predict future time series values. By using the QLoRA fine-tuning method [211], LLM-ABBA strikes a balance between task performance and efficiency, ensuring adaptability across various domains. As a result, LLMs are able to incorporate the COP of time series and analyze them from a macroscopic perspective, supported by domain knowledge from instructive prompts.

Our contributions include:

1. We propose a unified and improved ABBA approach for efficiently symbolizing multiple time series and mitigating accumulated shifts in time series reconstruction, facilitating effective inference over out-of-sample data.

2. For time series regression tasks, LLM-ABBA achieves SOTA performance, and it also delivers competitive results in medical time series classification tasks. To the best of our knowledge, this is the first work to practically integrate LLMs with STSA, and we believe our approach can be easily extended to other STSA methods.
3. LLM-ABBA retains language semantics and learns the COPs of time series via adapter fine-tuning methods in time series forecasting tasks.
4. The multi-modality and universality of LLMs in time series tasks lead to significant improvements.

The rest of the paper is structured as follows. Section 6.2 discusses related work in applications of LLMs to time series. Section 6.3 lays the foundation of the ABBA method and proposes our LLM-ABBA framework. Section 6.4 presents the simulations of our method as well as the comparisons between our method and SOTA methods. Section 6.5 discusses the limitations of our method and future work. Section 6.6 concludes the paper.

6.2 Related work

LLMs for time series methods have seen significant advancements in recent years. The work by [205] suggests that this success arises from the ability of LLMs to naturally model multimodal distributions of time series data. By framing time series forecasting as a sentence-to-sentence task, AutoTimes [212] minimizes the number of tunable parameters needed to generate time series embeddings, while freezing the parameters of the LLM. On the other hand, Frozen Pretrained Transformer (FPT) [68] fine-tunes LLM parameters to serve as a general representation extractor for various time series analysis tasks. These approaches capitalize on inherent token transitions, which improves model efficiency. For multivariate time series forecasting, UniTime [213] trains and fine-tunes a language model to offer a unified forecasting framework across multiple time series domains. Using advanced prompting techniques, PromptCast [214] converts time series data into text pairs, while TEMPO [215] models specific time series patterns—such as trends and seasonality—using weighted scatterplot smoothing [216].

Tuning-based predictors leverage accessible LLM parameters, typically involving pre-processing and tokenizing numerical signals alongside related prompt text, followed by fine-tuning on time series tasks [202]. To summarize, there are four key steps required to adapt LLMs to time series tasks:

1. $\mathcal{T}_{\text{inp}} = \text{Pre-processing}(\mathcal{T})$: The time series set \mathcal{T} is pre-processed to generate specific knowledge-contained inputs \mathcal{T}_{inp} , using operations such as a patching operation [203], [212] or weighted scatterplot smoothing [215];

2. $\mathcal{M}_{\text{inp}} = \text{Tokenizer}(\text{Prompt}, \mathcal{T}_{\text{inp}})$: An optional step involves performing a tokenizer operation on the time series \mathcal{T}_{inp} and the related prompt text to create a sequence of text tokens \mathcal{M}_{inp} ;
3. $\mathcal{M}_{\text{outp}} = f_{\text{LLM}}^{\Delta}(\mathcal{M}_{\text{inp}})$: With the instruction prompt **Prompt**, time series tokens (and any optional text tokens) are fed into $f_{\text{LLM}}^{\Delta}(\cdot)$, where partial unfreezing or additional adapter layers may be used. The output $\mathcal{M}_{\text{outp}}$ can either be a fine-tuned result or an intermediate result;
4. $\hat{Y} = \text{Task}(\mathcal{M}_{\text{outp}})$: Finally, an additional task operation, denoted as $\text{Task}(\cdot)$, is applied to generate the required output label \hat{Y} for the specific analysis task.

Algorithm 2: Greedy sorting-based aggregation

1. Scale and sort data points, and assume they are denoted p_1^s, \dots, p_n^s . Label all of them as “unassigned”.
 2. For $i \in \{1, \dots, n\}$ let the first unassigned point p_i^s as *starting point* and set $j := i$. If there are no unassigned points left, go to Step 6.
 3. If $\|p_i^s - p_j^s\|_2 \leq \alpha$,
 - assign p_j^s to the same group as p_i^s
 - increase $j := j + 1$
 4. If $j > n$ or termination condition is satisfied, go to Step 2. Otherwise go to Step 3.
 5. For each group computed, compute the center of the group as the mean of all its points.
-

6.3 Methodologies

6.3.1 ABBA Symbolic Approximation

Our approach is inspired by the observation that speech signals often contain rich semantic information [217], enabling language models to excel across a wide range of tasks [202] and references therein. However, directly applying language models to time series data is challenging due to the numerical nature of time series and the absence of useful embedding patterns. Additionally, the high dimensionality of time series makes it difficult for sequential or recurrent models to capture dependencies within the data. As such, learning a symbolic representation of time series while reducing their dimensionality presents a

practical yet complex problem. The [ABBA](#) method, a symbolic approximation approach, addresses this challenge by compressing the time series into a symbolic representation that encodes amplitude and period, with each symbol reflecting the oscillatory behavior of the time series over a specific period.

[ABBA](#) utilizes adaptive polygonal chain approximation followed by mean-based clustering to symbolically represent time series. The reconstruction error associated with this representation can be modeled as a *Brownian bridge* with pinned start and end points. [ABBA](#) symbolization involves two primary procedures: *compression* and *digitization*, which together aggregate a time series $T = [t_1, t_2, \dots, t_n] \in \mathbb{R}^n$ into its symbolic representation $A = a_1 a_2 \dots a_N$, where $N \ll n$ and a_i is an element from a specific letter set \mathcal{L} , known as the *dictionary* in the [ABBA](#) procedure.

Compression

The [ABBA](#) compression step calculates an [Adaptive Piecewise linear Continuous Approximation \(APCA\)](#) of the time series T . Compression plays a critical role in dimensionality reduction, where a user-defined tolerance, denoted `tol`, determines the degree of reduction. The compression process begins by adaptively selecting $N + 1$ indices $i_0 = 0 < i_1 < \dots < i_N = n$, such that the time series T is approximated well by a polygonal chain passing through the points (i_j, t_{i_j}) for $j = 0, 1, \dots, N$. This results in a partition of T into N segments $p_j = (\text{len}_j, \text{inc}_j)$, representing the cardinality and increment of each subseries $T_{i_{j-1}:i_j} = [t_{i_{j-1}}, t_{i_{j-1}+1}, \dots, t_{i_j}]$, where $\text{len}_j \in \mathbb{N}$ is the segment length, and $\text{inc}_j \in \mathbb{R}$ is the increment. Each segment p_j is represented by a straight line connecting the endpoints $t_{i_{j-1}}$ and t_{i_j} .

The partitioning criterion for selecting indices ensures that the squared Euclidean distance of the values in each segment p_j from the straight line is bounded by $(\text{len}_j - 1) \cdot \text{tol}^2$. Mathematically, this criterion is expressed as:

$$\sum_{i=i_{j-1}}^{i_j} \left(t_{i_{j-1}} + (t_{i_j} - t_{i_{j-1}}) \cdot \frac{i - i_{j-1}}{i_j - i_{j-1}} - t_i \right)^2 \leq (i_j - i_{j-1} - 1) \cdot \text{tol}^2. \quad (6.1)$$

Thus, the partitioning criterion ensures that the error in approximating each segment with a polygonal chain is bounded by $(\text{len}_j - 1) \cdot \text{tol}^2$. The polygonal chain can be recovered exactly from the initial time value t_0 and the tuple sequence $[p_1, p_2, \dots, p_N]$, with the reconstruction error modeled as a Brownian bridge. A smaller `tol` value ensures better compression, particularly for time series with complex features such as trends, seasonal cycles, and pulses. As noted in [71], the error bound between the original and reconstructed time series is upper-bounded by $(n - N) \cdot \text{tol}^2$.

Digitization

After compression, [ABBA](#) performs digitization to produce a symbolic representation. Prior to digitizing, the lengths and increments of the segments are normalized by their respective standard deviations, σ_{len} and σ_{inc} . Scaling is then applied using a parameter scl to control the relative importance of the segment length compared to its increment. The digitization process proceeds by clustering the scaled tuples $p_i^s = \left(\text{scl} \frac{\text{len}_i}{\sigma_{\text{len}}}, \frac{\text{inc}_i}{\sigma_{\text{inc}}} \right)$, where $i = 1, \dots, N$. If $\text{scl} = 0$, clustering is performed solely on the increment values, while if $\text{scl} = 1$, the lengths and increments are treated with equal importance.

The clustering procedure is based on a mean-based technique in Euclidean space. Given the scaled input $P^s = [p_1^s, \dots, p_N^s] \in \mathbb{R}^{2 \times N}$, the goal is to find a codebook $C = [c_1, \dots, c_k] \in \mathbb{R}^{2 \times k}$ of k clusters, where $k \ll N$, such that the [Sum of Squared Errors \(SSE\)](#) is minimized. A good codebook produces clusters S_1, S_2, \dots, S_k that minimize the [SSE](#), which is the sum of squared distances between each point $p^s \in S_i$ and its corresponding cluster center c_i .

In practice, the suboptimal k-means problem can be solved using a greedy sorting-based aggregation technique [72], which significantly speeds up the clustering process compared to traditional k-means. The clustering error is controlled by the parameter α , and the expected [SSE](#) value is given by $\frac{\alpha^2(N-k)}{2}$.

Once clustering is complete, each point p_i^s is assigned to the closest symbolic center $c^i \in C$, and the symbolic representation is constructed by assigning a unique symbol to each center. These symbols can be represented as text characters (e.g., ASCII codes or other character sets) and can be adapted to [LLM](#) pretrained tokens.

Inverse Symbolization

The inverse symbolization step converts the symbolic representation A back into a reconstructed time series \hat{T} , which is crucial for certain value prediction tasks. Inverse symbolization is followed by an *inverse-digitization* process, which uses the k representative cluster centers $c_i \in C$ to replace the symbols in A and denormalize them. This results in a 2-by- N array \tilde{P} , which is an approximation of P . Each $\tilde{p}_i \in \tilde{P}$ corresponds to the closest symbolic center $c^i \in C$.

Since the inverse digitization often leads to non-integer values for the reconstructed lengths len , a rounding method is applied to align the accumulated lengths with the nearest integers. The rounding is performed iteratively, starting with the first length:

$$(\widehat{\text{len}}_1, \widehat{\text{inc}}_1), (\widehat{\text{len}}_2, \widehat{\text{inc}}_2), \dots, (\widehat{\text{len}}_N, \widehat{\text{inc}}_N) \in \mathbb{R}^2, \quad (6.2)$$

After rounding, the final reconstructed time series \hat{T} is obtained by recovering \hat{P} from

the initial time value t_0 and the tuple sequence, as shown in Equation (6.2).

6.3.2 Error Analysis Reconstruction

We focus on the reconstruction error of ABBA's symbolization, as a symbolic representation with a higher reconstruction error is considered less informative. It is important to note that the reconstruction of time series from the compression procedure is achieved by forming a polygonal chain \tilde{T} , which connects the selected tuples $\{(i_j, t_{i_j})\}_{j=0}^N$ from the original time series T with $len_j = i_{j+1} - i_j$. As described in [71], the polygonal chain \hat{T} , formed by stitching together the tuples $\{(\hat{i}_j, \hat{t}_{i_j})\}_{j=0}^N$ via a tuple sequence \hat{P} , is reconstructed through inverse symbolization. This leads to Theorem 6.3.1.

Theorem 6.3.1. *Let $(\mu_i^{len}, \mu_i^{inc}) = \frac{1}{|S_i|} \sum_{(len, inc) \in S_i} (len, inc)$, where $\mathcal{U}_{len} = \{\mu_i^{len}\}_{i=1}^k$ and $\mathcal{U}_{inc} = \{\mu_i^{inc}\}_{i=1}^k$ represent the mean sets for **len** and **inc**, respectively. Since $i_0 = 0$, the reconstruction indices and size of time series values are given by:*

$$(\hat{i}_j, \hat{t}_{i_j}) = \left(\sum_{\ell=1}^j \widehat{len}_\ell, t_0 + \sum_{\ell=1}^j \widehat{inc}_\ell \right), \quad \text{for } j = 0, \dots, N, \quad (6.3)$$

where $(\widehat{len}_\ell, \widehat{inc}_\ell)$ are the computed cluster centers, i.e., $\widehat{len}_\ell \in \mathcal{U}_{len}$ and $\widehat{inc}_\ell \in \mathcal{U}_{inc}$.

Theorem 6.3.1 demonstrates that the accumulated deviations from the true lengths and increments are canceled out (as analyzed in [71]) at the right endpoint of the last piece p_N . Thus, $(\hat{i}_N, \hat{t}_{i_N}) = (i_N, t_{i_N}) = (n, t_n)$, indicating that the start and end points of \hat{T} , \tilde{T} , and T are identical. Consequently, we obtain the following result.

We define the local deviation of the increment and length as:

$$d_\ell^{inc} := \widehat{inc}_\ell - \widetilde{inc}_\ell, \quad d_\ell^{len} := \widehat{len}_\ell - \widetilde{len}_\ell. \quad (6.4)$$

Theorem 6.3.2 ([71]).

$$\sum_i \sum_{(len, inc) \in S_i} (d_\ell^{len}, d_\ell^{inc}) = (0, 0).$$

Theorem 6.3.3. *Assume that ABBA is performed with hyperparameter α , resulting in k clusters S_1, \dots, S_k . Then, we have:*

$$\max_\ell \{(d_\ell^{inc})^2 + (d_\ell^{len})^2\} \leq \alpha^2, \quad (6.5)$$

and further:

$$\sigma = \max_{i=1, \dots, k} \frac{1}{|S_i|} \sum_{(len, inc) \in S_i} \left(|len - \mu_i^{len}|^2 + |inc - \mu_i^{inc}|^2 \right) \leq \alpha^2,$$

Following Theorem 6.3.3, σ is explicitly controlled by α , eliminating the need to estimate the additional parameter tol_s used in [71], as it is now directly related to the hyperparameter α .

Given the N data points selected by the adaptive polygonal approximation chain, we define: $e_j^{\text{len}} := \sum_{\ell=1}^j d_\ell^{\text{len}}$ and $e_j^{\text{inc}} := \sum_{\ell=1}^j d_\ell^{\text{inc}}$. It is evident that $e_j^{\text{inc}} = \widehat{t}_{i_j} - t_{i_j}$ if $e_j^{\text{len}} = 0$ for $j = 1, \dots, N$. This leads to Theorems 6.3.4 and 6.3.5.

Theorem 6.3.4.

$$|e_j^{\text{inc}}| \leq j \sqrt{\alpha^2 - (d_\ell^{\text{len}})^2} \leq j|\alpha|,$$

where $j = 0, \dots, N$.

Similarly, the shift of the time series satisfies $|e_j^{\text{len}}| \leq j \sqrt{\alpha^2 - (d_\ell^{\text{inc}})^2} \leq j|\alpha|$ for $j = 0, \dots, N$.

Theorem 6.3.5.

$$\mathbb{P}(|e_j^{\text{inc}}| \geq h) \leq \exp\left(-\frac{h^2}{2j\alpha^2}\right) \quad \text{and} \quad \mathbb{P}(|e_j^{\text{len}}| \geq h) \leq \exp\left(-\frac{h^2}{2j\alpha^2}\right).$$

for all $h > 0$.

Proof 6.3.5.1 (Proof of Theorem 6.3.5). From Theorem 6.3.2, we obtain:

$$(e_0^{\text{len}}, e_0^{\text{inc}}) = (0, 0), \quad (e_N^{\text{len}}, e_N^{\text{inc}}) = (0, 0)$$

with expectation $E(e_j^{\text{len}}) = E(e_j^{\text{inc}}) = 0$.

For $j = 1, \dots, N$, since $d_j^{\text{len}}, d_j^{\text{inc}} \in [-\alpha, \alpha]$, using Equation (6.5) and Hoeffding's inequality, we get:

$$\begin{aligned} \mathbb{P}\left(\left|\sum_{\ell=1}^j (d_\ell^{\text{inc}} - E[d_\ell^{\text{inc}}])\right| \geq h\right) &= \mathbb{P}(|e_j^{\text{inc}} - E[e_j^{\text{inc}}]| \geq h) \\ &\leq \exp\left(-\frac{h^2}{2j\alpha^2}\right). \end{aligned}$$

Thus, $\mathbb{P}(|e_j^{\text{len}}| \geq h) \leq \exp\left(-\frac{h^2}{2j\alpha^2}\right)$ and $\mathbb{P}(|e_j^{\text{inc}}| \geq h) \leq \exp\left(-\frac{h^2}{2j\alpha^2}\right)$ for all $h > 0$.

This implies that a decrease in α tends to result in a smaller reconstruction error e_j . As noted in [71], the growth of j increases the likelihood of larger errors, as errors from previous reconstructions accumulate into subsequent ones through the inverse symbolization process.

6.3.3 ABBA to LLM

In this section, we define a single time series containing n data points as T , and let $\mathcal{T} = \{T_i\}_{i=1}^q$ represent a set of q time series, with the corresponding symbolic representation set $\mathcal{A} = \{A_i\}_{i=1}^q$.

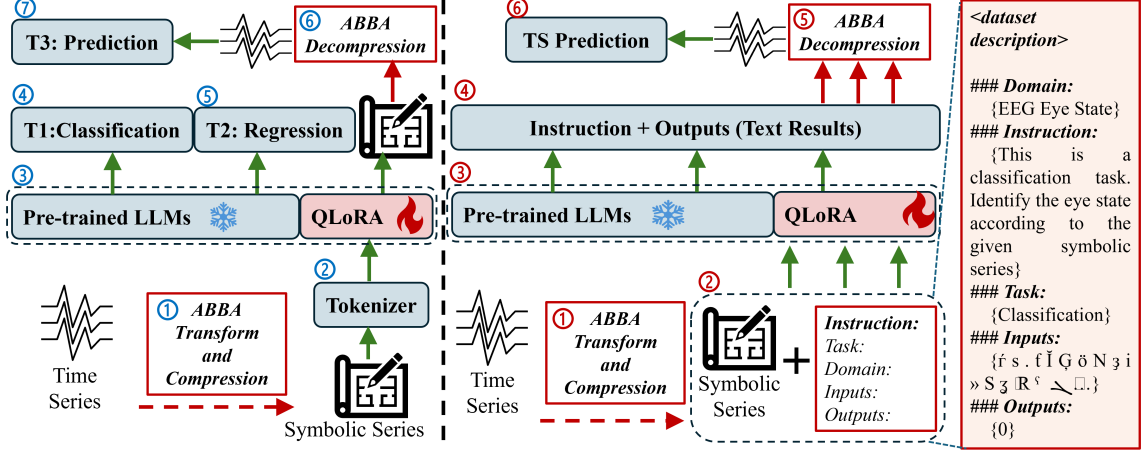


Figure 6.3: The LLM-ABBA framework: Given an input time series, we first transform and compress the time series into a symbolic series via steps ① and ①. These symbolic series are then tokenized using the LLM’s tokenizer ②. The instruction containing the symbolic series is also tokenized by the LLM’s tokenizer ②. By fine-tuning the pretrained LLM, the QLoRA with inhibition mechanism is applied in both ③ and ③. To implement the corresponding tasks, ④ and ⑤ load the LLM according to the task type, while ④ loads the LLM for generation tasks. For symbolic series inversion, ⑥ and ⑤ use ABBA to decompress the generated symbolic series. Finally, in ⑦ and ⑥, the output time series from LLM-ABBA is projected to generate forecasts.

Fixed-point Adaptive Polygonal Chain

In time series prediction settings, value-based prediction is converted into token-based prediction using STSA. However, it is desirable to mitigate the negative effect of previously predicted symbols on subsequent time series recovery, as the recovery proceeds from front to back. APCA and symbolic recovery often lead to cumulative errors in symbolic prediction, meaning that an incorrect symbol from earlier in the sequence will affect the reconstruction of subsequent symbols. To address this, a fixed-point polygonal chain technique is introduced.

We partition the time series into segments following Equation (6.1), where $p_j = (\text{len}_j, \text{inc}_j)$ is replaced with $p_j = (\text{len}_j, t_{i_j})$ before normalization. This new approximation method is called Fixed-point Adaptive Piecewise linear Continuous Approximation (FAPCA). The resulting tuples p_i are normalized, and since $\text{inc}_j = t_{i_j} - t_{i_{j-1}}$, they can be recovered from each other. Figure 6.4 shows that FAPCA eliminates cumula-

tive errors from earlier mistaken symbols, improving recovery. ABBA with APCA and FAPCA generate symbols “aBbBbBbBbBbBbBA” and “abBbBbBbBbBbBbBA”, respectively, along with their respective perturbed symbols, “abbBbBbBbBbBbBA” and “aBBbBbBbBbBbBbBA”. Symbol recovery is performed on both the correct and perturbed symbols.

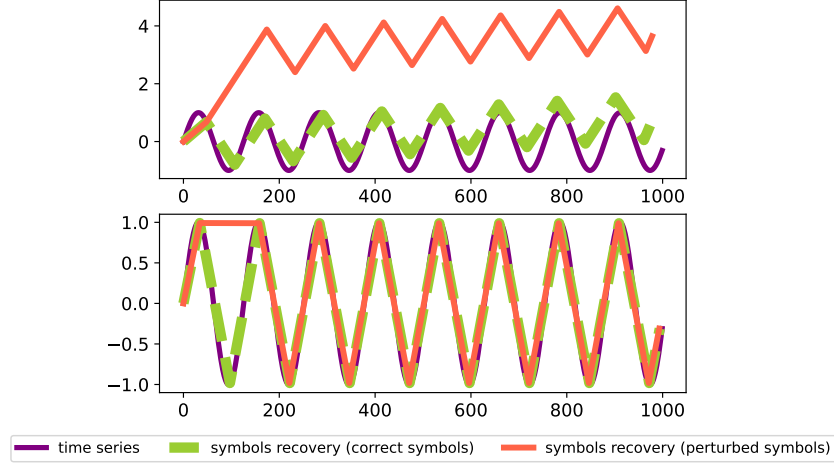


Figure 6.4: A synthetic trigonometric sine series with 1,000 points is generated, and symbolic approximation using 4 symbols is separately performed with APCA (the upper panel) and FAPCA (the lower panel) on the time series.

Symbolizing Multiple Time Series

Existing symbolic approximation methods focus on converting a single time series. However, they are not designed to handle multiple time series with consistent symbolic information, where each symbol corresponds to a unique symbolic center. To manage co-evolving or multiple time series, it is necessary to maintain consistent symbolic information across different symbolic representations.

We propose a unified approach to achieve consistent symbolic approximation for multiple time series:

- Step 1: Use [APCA](#) or [FAPCA](#) to compress each time series T_i into P_i for $i = 1, \dots, q$.
- Step 2: Compute the normalized P_i^s for each series and concatenate these to form $\mathcal{P}^s := [P_i^s]_{i=1}^q$.
- Step 3: Perform digitization on \mathcal{P}^s .
- Step 4: Allocate symbols to each time series, where the number of symbols for T_i is equal to $|P_i^s|$.

Symbolizing Out-of-Sample Data

To symbolize out-of-sample time series data with consistent symbols, which is essential for downstream tasks like inference, we follow these steps:

- Step 1: Compress each time series T_i^t into P_i^t for $i = 1, \dots, q'$.
- Step 2: Assign a symbol to each $p \in P_i^t$ following the digitization rule.

Feeding the LLM

ABBA transforms numerical time series into symbolic series while preserving the internal logic chain that can be learned by LLMs. By ensuring the symbolic series inherits the polygonal chain of the original time series and represents it via tokens, LLMs can reconstruct the embedding space without requiring new tokens, by adapting fine-tuning methods.

As shown in Figure 6.3, the left panel represents the traditional setup for tasks like classification, regression, and prediction, while the right panel corresponds to the instruction-based setup. Instructions (the right panel) guide the LLMs to understand the tasks, making it equivalent to the left panel (without LLMs' instructions) in terms of task execution.

For the consistency of tuning-based methods, let \mathcal{T} represent the input time series dataset and \mathcal{A} the symbolic representation generated by ABBA. The symbolization of ABBA is denoted by $\phi : \mathcal{T} \rightarrow \mathcal{A}$, and its inverse symbolization is denoted by $\phi^{-1} : \mathcal{A} \rightarrow \mathcal{T}$. We define the LLM-ABBA framework as follows:

1. $\mathcal{A} = \phi(\mathcal{T})$: The input \mathcal{T} is converted to its symbolic representation \mathcal{A} .
2. $\mathcal{M}_{\text{inp}} = \text{Tokenizer}(\text{Prompt}, \mathcal{A})$: Tokenize the symbolic representation \mathcal{A} using the LLM's default tokenizer.
3. $\mathcal{M}_{\text{outp}} = f_{\text{LLM}}^{\Delta}(\mathcal{M}_{\text{inp}})$: Feed the tokenized input into the LLM model.
4. $\hat{Y} = \text{Task}(\mathcal{M}_{\text{outp}})$: Depending on the task type:

$$\begin{cases} \hat{Y} = \mathcal{M}_{\text{outp}}, & \text{Classification task,} \\ \hat{Y} = \phi^{-1}(\mathcal{M}_{\text{outp}}), & \text{Regression / Prediction task} \end{cases}$$

6.3.4 Linguistic Investigation: Zipf's Law

In most corpora, the most frequent word appears approximately twice as often as the second most frequent word; this phenomenon is described by Zipf's law [218]. Zipf's law

asserts that the frequencies of certain events are inversely proportional to their rank, and the rank-frequency distribution follows an inverse power law.

In Figure 6.5, we observe that the unigrams generated by ABBA symbolization from seven different time series datasets from the UCR Archive roughly adhere to Zipf’s law. This illustrates an interesting alignment between ABBA symbols and the distribution of words in natural language.

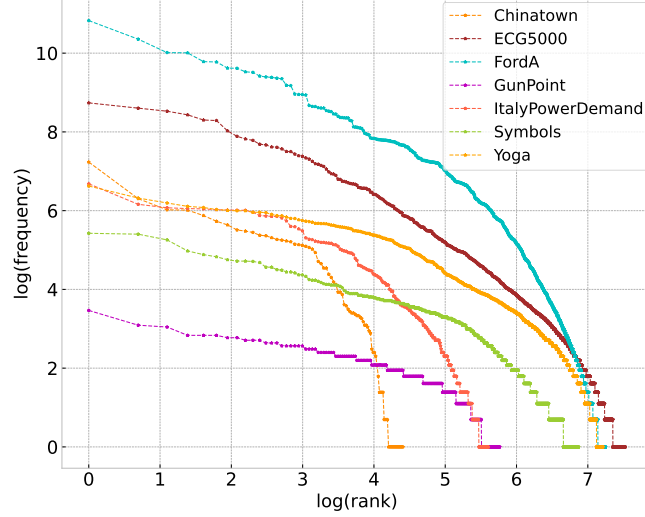


Figure 6.5: Frequency and rank of symbols in various UCR datasets.

6.4 Experiments

In this section, we explore three time series tasks to evaluate the effectiveness of ABBA within LLM. Additionally, we fine-tune three language models on the training data using QLoRA [211] with inhibition [219]. All experiments are conducted in PyTorch on a single NVIDIA A100 40GB GPU. The advantages of LLM-ABBA include (1) eliminating the need for LLMs to learn time series data from scratch, and (2) relying solely on compression and decompression, without the need for training additional embedding layers [204]. For a fair comparison, we evaluate all models under the same settings for each task. Unless otherwise specified, we assume that greedy aggregation is used for the ABBA digitization.

A larger dataset requires more symbols or LLM tokens because it contains more information and symbolic semantics. RoBERTa-Large, based on BERT [156], processes input sentences bidirectionally, while Llama2-7B and Mistral-7B, which originate from the GPT architecture [157], operate unidirectionally (from left to right). Causality analysis, commonly used to compute the context of multichannel EEG signals, is also applicable to medical time series analysis. However, electrocardiogram (ECG) signals typically rely on sequential features. Therefore, when using LLM-ABBA for medical time series analysis, it

is crucial to first consider the properties and characteristics of the data. In some cases, we were unable to reproduce or find [SOTA](#) performance numbers. For a comprehensive analysis, we test [ABBA](#) with [LLMs](#) on three core time series analysis tasks. Three [LLMs](#) are used to process the [COP](#) in symbolic series: M1 (RoBERTa_{Large}) [160], M2 (Llama2-7B) [220], and M3 (Mistral-7B) [221].

6.4.1 Hyperparameters

Hyperparameters of ABBA

There are four key parameters that govern the transition of time series when integrating [ABBA](#) into [LLMs](#). The tolerance parameter `tol` is selected from $\{1 \times 10^{-2}, 1 \times 10^{-4}, 1 \times 10^{-6}\}$ to control the degree of compression and dimensionality reduction. The digitization parameter α is chosen from $\{1 \times 10^{-2}, 1 \times 10^{-4}, 1 \times 10^{-6}\}$ to determine the number of distinct symbols. \mathcal{L} is a finite letter set that specifies the tokens used by [LLMs](#), and `sc1` $\in \{1, 2, 3\}$ is used as a normalized scaling factor for the length of each segment.

Hyperparameters of LLMs

Table 6.1: Hyperparameters of Classification tasks. Quant. is the model quantization process. Inhib. is the inhibition threshold in QLoRA. Embed. means to save tuned embeddings. Optim. is the optimization method. LR is the learning rate. Acc. is the accuracy rate (%).

LLM-ABBA on Classification Tasks												
Models	Quant. Tokens		Metric	LoRA					Optim.	Epochs LR		Batch Size
	4-bits	Length		alpha	low	rank	r	dropout		inhib.	Embed.	
RoBERTa _{Large}	True	512	Acc.	16	16, 64, 256	0.05	0.3	Save	adamw_8bit	10	5e-7	4
Llama2-7B	True	4,096	Acc.	16	16, 64, 256	0.05	0.3	Save	adamw_8bit	10	5e-7	4
Mistral-7B	True	4,096	Acc.	16	16, 64, 256	0.05	0.3	Save	adamw_8bit	10	5e-7	4

Table 6.2: Hyperparameters of Regression tasks. Quant. is the model quantization process. Inhib. is the inhibition threshold in QLoRA. Embed. means to save tuned embeddings. Optim. is the optimization method. RMSE is the root-mean-square-error.

LLM-ABBA on Regression Tasks												
Models	Quant. Tokens		Metric	LoRA					Optim.	Epochs	LR	Batch Size
	4-bit	Length		alpha	low	rank	r	dropout				
RoBERTa _{Large}	True	512	RMSE	16	16, 64, 256	0.05	0.3	Save	adamw_8bit	10	2×10^{-6}	4
Llama2-7B	True	4,096	RMSE	16	16, 64, 256	0.05	0.3	Save	adamw_8bit	10	2×10^{-6}	4
Mistral-7B	True	4,096	RMSE	16	16, 64, 256	0.05	0.3	Save	adamw_8bit	10	2×10^{-4}	4

There are three time series analysis tasks: classification, regression, and prediction. We quantize [LLMs](#) by 4-bits using the `bitsandbytes` package². In order to fine-tune [LLMs](#), the shunting inhibition mechanism [219] is utilized during the [QLoRA](#) adapter fine-tuning

²<https://github.com/bitsandbytes-foundation/bitsandbytes>

Table 6.3: Hyperparameters of Prediction tasks. Quant. is the model quantization process. Inhib. is the inhibition threshold in QLoRA. Embed. means to save tuned embeddings. Optim. is the optimization method. MAE is the mean-absolute-error, and MSE is the mean-square-error.

LLM-ABBA on Prediction Tasks												
Models	Quant. Tokens		Metric	LoRA					Optim.	Epochs	LR	Batch Size
	4-bit	Length		alpha	low rank	r	dropout	inhib.				
RoBERTa _{Large}	True	512	MAE, MSE	16	16, 64, 256	0.05	0.3	Save	adamw_8bit	10	2×10^{-6}	4
Llama2-7B	True	4,096	MAE, MSE	16	16, 64, 256	0.05	0.3	Save	adamw_8bit	10	2×10^{-6}	4
Mistral-7B	True	4,096	MAE, MSE	16	16, 64, 256	0.05	0.3	Save	adamw_8bit	10	2×10^{-6}	4

progress. The modified embedding layer is also saved after fine-tuning on the corresponding task. For the classification task, the metric is accuracy rate (%). Root-mean-square-error is used as the metric for regression tasks. Mean-square-error and mean-absolute-error are used as the metrics for prediction tasks, and we also visualize the correlation coefficient of prediction tasks on ETTh1 data in terms of their seven features. We control the fine-tuning epoch and apply a small batch size on every task. The alpha of QLoRA is set to 16.

6.4.2 Compression and Recovery

To transform the numerical time series to symbolic time series, we use tokens of LLMs as the initial dictionary of ABBA for the symbolic representation, and there are no extra tokens that will be used to represent the numerical input. ABBA shows a strong symbolic transition on time series signals (See Figure 6.6 and Table 6.4).

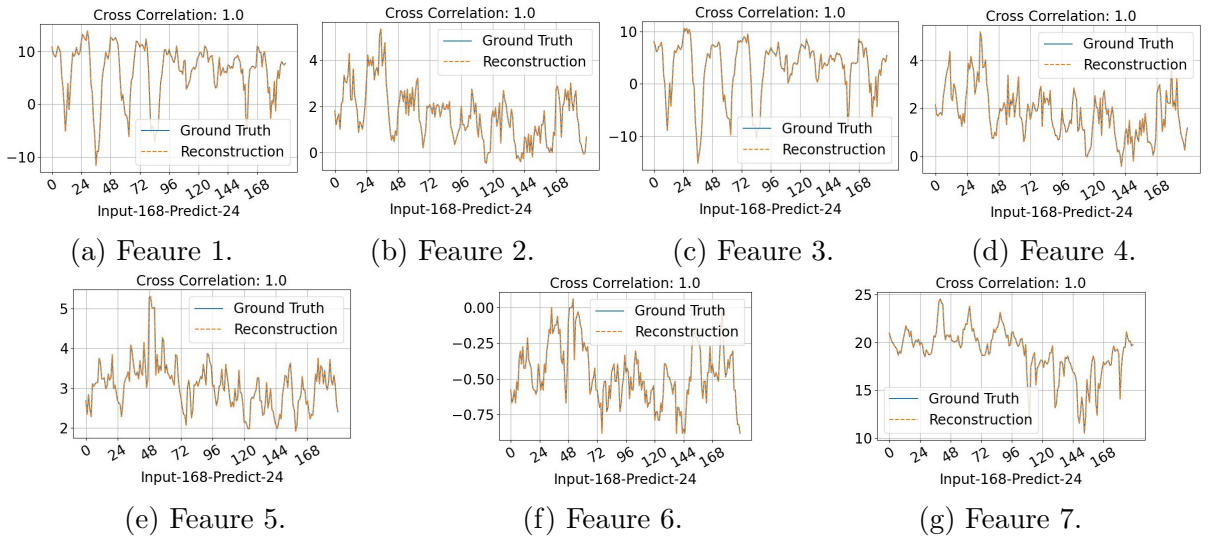


Figure 6.6: Visualization of reconstructed input-168-predict-24 results on ETTh1 data by using ABBA symbolic approximation, where $\text{tol} = 0.01$, $\alpha = 0.01$ and $\text{scl} = 3$.

To visualize the performance of ABBA on time series transition processes, we employ ETTh1 time series data to compute the correlation coefficient and reconstruction error of

ABBA. This multivariate data has seven features, and in terms of these seven features, the average of **Mean-Square Error (MSE)**, **Mean Absolute Error (MAE)**, and correlation coefficient between original time series input and reconstructed outputs is computed.

Table 6.4: Symbolic approximation performance on ETTh1 data using ABBA. ABBA describes a time series sample by using symbolic approximation, and the number of used symbols depends on the complexity of the data. If the time series sample is a regular wave (for example, a sine wave), the number of used symbols is small; otherwise, ABBA needs more symbols.

ABBA Settings		Number of Symbols	Reconstructed Time Series			
tol and α	scl		Used LLM's tokens	MSE	MAE	Correlation Coefficient
$1 \times 10^{-2}, 1 \times 10^{-2}$	3	846		2.5×10^{-7}	1×10^{-2}	1.0
$1 \times 10^{-4}, 1 \times 10^{-4}$	3	2,713		4.2×10^{-8}	1.4×10^{-4}	1.0
$1 \times 10^{-6}, 1 \times 10^{-6}$	3	2,789		3.2×10^{-8}	1.3×10^{-4}	1.0

In this section, we observe which **ABBA** settings better suit time series characteristics. The default **scl** is set to 3, which is used in other **LLM** tasks. **tol** and α are set to be the same. Table 6.4 reports the input-168-predict-96 results when using **ABBA** to reconstruct ETTh1 data in terms of seven features. Setting smaller **tol** and α in **ABBA** can reduce the **MSE** and **MAE** scores, but more symbols or **LLM** tokens will be used. Under all above conditions, the correlation coefficient is 1.0.

6.4.3 Time Series Classification Tasks

For the classification task, we evaluate these three pretrained **LLMs** on UCR Time Series Archive datasets [222], **EEG** eye state [223], and MIT-BIH [224], [225] which have been extensively adopted for benchmarking time series classification models. We utilize cross-entropy loss for the classification training. Details of the implementation and datasets can be found in Table 6.1. The evaluation metric is accuracy rate (%).

The UCR Archive contains 128 datasets already partitioned into train and test sets, although the ratio of the train set and test set is not always consistent³. These datasets have varying numbers of labels and feature dimensions. Also, there can be uneven numbers of labels, which often results in overfitting. Therefore, classifying time series in the UCR Archive is a challenging task. Table 6.5 reports the full time series classification results on UCR2018. J1 refers to results of using k-means in the digitization process, and J2 refers to the results of using greedy aggregation (Algorithm 2) in the digitization process. We

³The UCR Archive 2018 is available at https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

find that Algorithm 2 outperforms k-means symbolization time series transition progress in most cases.

In Table 6.5, we report the classification performance on a partial dataset of UCR2018. In most cases, although LLM-ABBA cannot outperform the SOTA in terms of time series classification tasks, ABBA with LLMs can reach an acceptable application requirement in some practical cases (such as “Coffee”, “Earthquakes”, “Herring”, “Strawberry”, “Trace”, “Wafer”, “WormsTwoClass”). Compared to V2S [226] which is the SOTA, although these three LLMs with the use of QLoRA occupies more memory, the multi-modality of LLMs, especially on time series analysis tasks, achieves a noticeable improvement.

In the medical domain (for example, identifying the eye state using EEG signals, distinguishing abnormal ECG signals, and classifying the “normal beats”, “supraventricular ectopy beats”, “ventricular ectopy beats”, “fusion beats”, and “unclassifiable beats” of ECG signals), we report the performance of LLM-ABBA on three medical time series datasets. We set $\text{tol} = \alpha = 0.01$. In Table 6.6, compared to CNN [227] on the PTB-DB data set, LLM-ABBA achieves performance almost equivalent to the SOTA. In the aspect of distinguishing MIT-BIH, CNN [227] and Bidirectional Recurrent Neural Networks (BiRNN) [224], [228] performs the best, but LLM-ABBA slightly outperforms LSTM [229], [230].

6.4.4 Time Series Regression Tasks

For the regression task, we evaluate these three pretrained LLMs on the TSER benchmarking archive [200], which contains 19 time series datasets from 5 application domains, including Health Monitoring, Energy Monitoring, Environment Monitoring, Sentiment Analysis, and Forecasting⁴. To use as few symbols as possible, we initialize the setting of $\text{tol} = 0.01$ and $\alpha = 0.01$. We also utilize the L2 loss for the regression training. Details of the implementation and datasets can be found in Table 6.2. The evaluation metric is RMSE.

Experimenting on the TSER benchmark archive [200], the empirical results are shown in Table 6.7, in which for 15 out of 19 use-cases, LLM-ABBA outperforms the machine learning SOTA results. We believe that LLM-ABBA can exploit the semantic information hiding beneath the time series in the task of time series regression. ABBA is able to provide COPs to LLMs by compressing and digitizing time series to symbols, which finally results in the change of embedding space by using adaption fine-tuning methods.

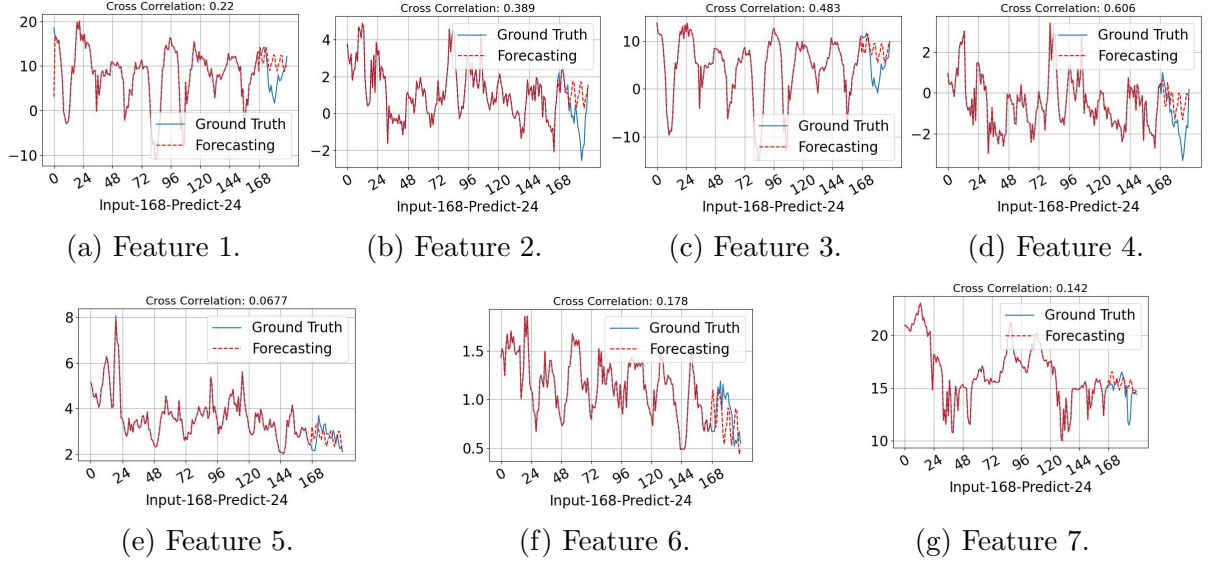


Figure 6.7: Visualization of input-168-predict-24 results on ETTh1 using LLM-ABBA.

6.4.5 Time Series Forecasting Tasks

For time series forecasting, we experimented on 4 well-established benchmarks: ETT datasets (including 4 subsets: ETTh1, ETTh2, ETTm1, ETTm2) [231], [232]. Details of the implementation and datasets can be found in Table 6.3. The input length of the time series is 168, and we use three different prediction horizons $H \in \{24, 96, 168\}$. The evaluation metrics include MSE and MAE.

Although LLM-ABBA cannot obtain a new SOTA on time series forecasting tasks, it compares favorably to the Informer architecture which is trained from scratch. The congenital defect of ABBA is that the symbolization tends to be affected by the fluctuation and oscillation of time series signals, which eventually leads to higher MSE and MAE scores. Because LLM-ABBA utilizes a totally different technical roadmap to existing methods, it only remolds the construction of the LLM’s tokens. However, remodeling pretrained tokens inevitably brings the previous pretrained semantics to the LLM-ABBA design. Thus, we discussed the semantic consistency of LLM-ABBA using extra symbols or tokens to overcome this problem.

6.4.6 QLoRA Fine-Tuning

Because the low rank of adapter fine-tuning will influence the efficiency of passing information [211], [219] from the previous layer, we use different low rank settings of QLoRA on the corresponding tasks during the fine-tuning progress. But for time series regression and prediction tasks, we select $r \in \{16, 46, 256\}$ for the corresponding data input. We

⁴Monash regression data is available at <http://tseregression.org/>.

find that there is no obvious over-fitting problem, and more tunable parameters are not able to improve the performance of LLM-ABBA.

In medical time series domains, ptb-db and MIT-BIH arrhythmia data sets are mostly used. EEG eye state data set has two categories, and because of its high complexity, the accuracy always stays at around 60%. EEG eye state data and MIT-BIH has more than one channel, which indicates that LLM-ABBA might have the ability to process complicate features across channels. Table 6.6 presents the full medical time series classification results using LLM-ABBA.

LLM-ABBA achieves comparable time series prediction results to the SOTAs, and there is no over-fitting in these tasks when using different low rank r . Because ABBA tends to symbolize trends and altitudes of the time series signals, LLM-ABBA always strengthens the vibration of predicted time series segments which can be seen in Figure 6.7.

6.4.7 Semantic consistency

When using pretrained tokens as the input symbols, fine-tuning on no language content (such as time series signals) will generally bring semantic loss to LLMs. Therefore, we use ASCII codes to generate new symbols by adding more digits and expanding the used alphabet table. Following the same fine-tuning process to the above experiment settings, we compute the forecasting performance by fine-tuning on Mistral-7B. Compared to Table 6.8, Table 6.9 shows that the difference is not noticeable.

6.5 Limitations

ABBA is evaluated through performance profiles based on its reconstruction, assessed using the 2-norm, Dynamic Time Warping (DTW), and their respective differenced measures. These evaluations show that ABBA performs competitively against SOTA methods, such as SAX. Previous STSA methods have been applied in various data mining applications, including EEG signal analysis [233] and the Internet of Things [234]. Additionally, ABBA demonstrates improved performance in anomaly detection tasks, such as TARZAN, by replacing SAX methods [71], [72].

LLMs are capable of understanding the generated symbols from ABBA. Each data sample is represented by symbols, with each symbol having a specific meaning that corresponds to a node in the internal COP of the time series data. LLM-ABBA excels not only in time series classification tasks but also in time series regression tasks (as shown in Table 6.5 and Table 6.7). Since these symbolic series follow a logical chain that reflects the trends in time series data, LLMs can learn the temporal patterns through adapter

fine-tuning methods. As illustrated in Figure 6.7, by applying the inverse symbolization process of ABBA, LLMs are able to predict time series trends with reduced drift in the forecasted segments. Therefore, time series forecasting tasks can also benefit from these findings. While ABBA effectively approximates time series through symbolic series, LLMs are prone to hallucinations, meaning that generating more content could lead to more “hallucinated” knowledge. As a result, LLM-ABBA tends to perform better on short-term time series prediction and regression tasks.

Our proposed FAPCA strategy for ABBA does not completely eliminate the potential for cumulative errors arising from incorrect symbols during recovery. A minor shift can occur if an incorrect len_i leads to improper symbol replacements. Furthermore, hallucination, which is an inherent issue with LLMs, is not fully addressed in this work. As a result, the vibration or adverse response of predicted sequences can negatively impact performance. Moreover, after using ABBA to transform time series data, most LLMs can only handle up to 4,096 tokens, which limits their capacity for long-term time series analysis.

6.6 Conclusion

In this paper, we introduce LLM-ABBA for time series classification, regression, and forecasting tasks. We discuss the seamless integration of time series symbolization with LLMs and highlight how this integration enhances performance. Theoretically, we analyze the reconstruction error of ABBA symbolization, its relationship with key parameters, and the inherent limitations of LLM-ABBA. To address the drift phenomenon in time series, we propose the FAPCA method, which improves ABBA symbolization. Empirical results demonstrate that our method achieves performance comparable to the SOTA in classification and regression tasks. In terms of convenience and universality, LLM-ABBA enhances the multi-modality of LLMs for time series analysis. We believe the potential of ABBA extends to other time series applications, which will be explored in future work.

Table 6.5: Full comparison of results for time series classification tasks(%) on UCR datasets.

Data	Classes Symbols		RoBERTa _{Large}			Llama2-7B			Mistral-7B			V2Sa [226]	
	Number	Number	Para.	J1	J2	Para.	J1	J2	Para.	J1	J2	Para.	SOTA
BME	3	836	2.65M	34.0	60.2	12.7M	41.3	84.7	9.56M	43.3	77.3	0.3M	-
BeetleFly	2	731	2.65M	65.0	95.0	12.7M	50.0	65.0	9.56M	55.0	75.0	0.3M	-
BirdChicken	2	424	2.65M	55.0	70.0	12.7M	60.0	65.0	9.56M	55.0	75.0	0.3M	-
ChinaTown	2	585	2.65M	72.0	72.6	12.7M	58.3	84.3	9.56M	61.5	89.2	0.3M	-
Coffee	2	701	2.65M	50.0	89.3	12.7M	60.7	96.5	9.56M	78.6	89.3	0.3M	100
DistalPhalanxOutlineAgeGroup	3	1,444	2.65M	68.3	68.3	12.7M	71.2	73.4	9.56M	67.6	74.8	0.3M	-
DodgerLoopWeekend	2	143	2.65M	72.6	73.9	12.7M	70.3	64.5	9.56M	69.6	71.7	0.3M	-
ECG200	2	1,781	2.65M	70.0	68.0	12.7M	63.0	64.0	9.56M	66.8	68.0	0.3M	87.4
ECG5000	5	10,334	2.65M	81.2	76.0	12.7M	75.7	74.7	9.56M	75.4	73.4	0.3M	94.0
ECGFiveDays	2	2,463	2.65M	52.6	56.9	12.7M	53.3	63.9	9.56M	49.5	68.8	0.3M	-
Earthquakes	2	940	2.65M	52.7	74.8	12.7M	77.7	76.3	9.56M	79.1	76.3	0.3M	78.4
FordA	2	9,759	2.65M	68.9	68.9	12.7M	58.7	61.1	9.56M	62.7	60.9	0.3M	100
FordB	2	9,352	2.65M	68.9	58.1	12.7M	56.1	58.9	9.56M	55.1	57.0	0.3M	100
FreezerRegularTrain	2	2,663	2.65M	61.9	74.5	12.7M	64.1	76.1	9.56M	63.2	75.4	0.3M	-
FreezerSmallTrain	2	2,593	2.65M	62.3	74.1	12.7M	63.8	67.8	9.56M	63.3	67.5	0.3M	-
GunPoint	2	791	2.65M	51.4	73.3	12.7M	54.0	82.7	9.56M	48.0	80.0	0.3M	96.7
GunPointAgeSpan	2	2,057	2.65M	83.5	94.3	12.7M	69.9	84.5	9.56M	67.1	85.5	0.3M	-
GunPointMaleVersusFemale	2	2,057	2.65M	57.9	76.3	12.7M	59.8	71.2	9.56M	55.7	74.1	0.3M	-
GunPointOldVersusYoung	2	2,057	2.65M	66.7	97.5	12.7M	62.9	85.1	9.56M	67.9	80.0	0.3M	-
HandOutlines	2	7,572	2.65M	66.5	77.0	12.7M	63.5	68.6	9.56M	65.1	71.6	0.3M	93.2
Herring	2	982	2.65M	59.4	65.6	12.7M	62.5	62.5	9.56M	54.7	60.9	0.3M	68.8
HouseTwenty	2	1,385	2.65M	50.8	67.1	12.7M	69.7	89.1	9.56M	75.6	93.3	0.3M	-
ItalyPowerDemand	2	1,759	2.65M	59.7	70.4	12.7M	55.7	73.4	9.56M	53.4	73.2	0.3M	97.1
Lightning2	2	2,175	2.65M	67.2	65.6	12.7M	68.9	65.6	9.56M	67.2	62.3	0.3M	100
Meat	3	161	2.65M	55.0	70.0	12.7M	68.3	70.0	9.56M	66.7	70.0	0.3M	-
MelbournePedestrian	10	1,081	2.65M	34.6	68.5	12.7M	27.1	76.8	9.56M	29.2	74.4	0.3M	-
MiddlePhalanxOutlineCorrect	2	1,700	2.65M	59.8	67.4	12.7M	58.1	69.8	9.56M	61.2	67.7	0.3M	91.1
MiddlePhalanxTW	6	1345	2.65M	53.9	54.5	12.7M	53.9	48.7	9.56M	51.9	46.8	0.3M	84.9
OliveOil	4	150	2.65M	66.7	46.7	12.7M	76.7	70.0	9.56M	73.3	73.3	0.3M	-
PhalangesOutlinesCorrect	2	2,785	2.65M	62.2	65.4	12.7M	63.9	67.5	9.56M	62.7	67.5	0.3M	-
Plane	7	1,424	2.65M	33.3	81.0	12.7M	39.0	78.1	9.56M	38.1	83.8	0.3M	-
PowerCons	2	2,007	2.65M	77.8	79.0	12.7M	72.8	81.1	9.56M	77.8	80.6	0.3M	-
ProximalPhalanxOutlineCorrect	2	1,298	2.65M	71.5	82.8	12.7M	73.9	85.6	9.56M	72.9	83.9	0.3M	-
ProximalPhalanxTW	6	1,101	2.65M	67.8	80.0	12.7M	69.8	80.0	9.56M	68.8	74.1	0.3M	-
SemgHandGenderCh2	4	2,840	2.65M	49.1	54.7	12.7M	59.5	67.2	9.56M	58.3	73.3	0.3M	-
SmallKitchenAppliances	2	2,207	2.65M	66.2	69.3	12.7M	60.8	63.2	9.56M	57.6	61.6	0.3M	83.5
SonyAIBORobotSurface1	2	2,558	2.65M	54.2	60.4	12.7M	64.1	71.7	9.56M	68.2	78.5	0.3M	-
StarLightCurves	3	27,131	2.65M	67.8	72.9	12.7M	68.6	72.6	9.56M	67.6	70.1	0.3M	-
Strawberry	2	3,593	2.65M	71.2	85.1	12.7M	69.5	84.9	9.56M	69.5	88.4	0.3M	97.6
ToeSegmentation2	2	2,714	2.65M	79.7	73.1	12.7M	69.2	59.2	9.56M	77.7	80.0	0.3M	-
Trace	4	870	2.65M	49.5	88.0	12.7M	54.0	90.0	9.56M	47.0	77.0	0.3M	100
TwoLeadECG	2	2,487	2.65M	59.6	69.1	12.7M	53.2	64.6	9.56M	53.2	63.9	0.3M	97.8
Wafer	2	4,805	2.65M	94.6	96.8	12.7M	91.3	93.5	9.56M	90.9	95.2	0.3M	100
Wine	2	171	2.65M	53.6	57.4	12.7M	59.3	63.0	9.56M	63.0	55.6	0.3M	90.7
Worms	5	5,377	2.65M	62.6	67.5	12.7M	57.1	64.9	9.56M	54.5	63.6	0.3M	83.1
WormsTwoClass	2	5377	2.65M	74.3	81.8	12.7M	62.3	70.1	9.56M	61.0	79.2	0.3M	98.7

Table 6.6: Full comparison of results on medical time series classification tasks(%) on EEG eye states, ptb-db, and MIT-BIH.

Data	Classes	Symbols	RoBERTa _{Large}			Llama2-7B			Mistral-7B			CNN	BiRNN	LSTM
			r=16	r=64	r=256	r=16	r=64	r=256	r=16	r=64	r=256	[227]	[228]	[230]
EEG	2	938	60.1	66.0	64.4	55.9	57.4	57.5	58.5	58.0	60.1	53.1	55.3	50.7
ptb-db	2	2,179	89.5	90.6	89.3	99.0	98.6	98.3	98.9	98.7	98.6	99.4	97.0	90.7
mit-bih	5	2,926	86.4	86.4	86.3	89.6	89.4	89.1	89.3	89.7	89.3	93.4	96.5	88.1

Table 6.7: Full comparison of results on the regression task on 19 Monash Time Series Regression datasets.

Data	Symbols Number	RoBERTa _{Large}			Llama2-7B			Mistreal-7B			SOTA
		r=16	r=64	r=256	r=16	r=64	r=256	r=16	r=64	r=256	[200]
		RMSE	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE
AppliancesEnergy	778	1.73	2.09	1.74	2.43	2.43	2.43	2.34	2.02	2.11	2.29
HouseholdPowerConsumption1	1717	377.02	377.20	377.20	398.01	398.05	398.05	228.83	228.78	228.67	132.80
HouseholdPowerConsumption2	1717	27.64	27.71	27.73	36.63	36.71	36.69	24.54	24.56	24.51	32.61
BenzeneConcentration	3037	4.01	4.00	4.00	5.57	5.56	5.56	4.03	4.03	4.03	0.64
BeijingPM10Quality	970	66.16	66.07	66.07	93.25	93.26	93.26	65.25	65.25	65.24	93.14
BeijingPM25Quality	970	54.16	54.16	54.16	76.75	76.73	76.73	53.50	53.49	53.49	59.50
LiveFuelMoistureContent	5689	20.56	20.56	20.56	29.32	29.33	29.32	20.94	20.88	20.85	29.41
FloodModeling1	969	0.00	0.00	0.00	0.05	0.05	0.05	0.37	0.36	0.36	0.00
FloodModeling2	979	0.00	0.00	0.00	0.05	0.04	0.04	0.40	0.39	0.39	0.01
FloodModeling3	948	0.00	0.00	0.00	0.06	0.05	0.05	0.41	0.37	0.39	0.00
AustraliaRainfall	4740	4.36	4.36	4.36	6.05	6.01	6.02	4.31	4.28	4.30	8.12
PPGDalia	12298	9.32	9.32	9.32	12.54	12.50	12.52	9.04	9.02	9.03	9.92
IEEEPPG	8971	17.06	17.00	17.04	22.59	22.53	22.55	17.15	17.12	17.16	23.90
BIDMC32HR	9423	6.73	6.98	6.71	12.02	11.98	12.04	8.24	8.21	8.23	9.42
BIDMC32RR	9412	1.77	1.74	1.76	2.64	2.61	2.62	2.09	2.06	2.08	3.02
BIDMC32SpO2	5537	2.90	2.85	2.89	3.82	3.79	3.81	2.95	2.91	2.93	4.45
NewsHeadlineSentiment	5537	0.07	0.07	0.07	0.13	0.13	0.13	0.11	0.11	0.11	0.14
NewsTitleSentiment	5537	0.07	0.07	0.07	0.13	0.13	0.13	0.11	0.11	0.11	0.14
Covid3Month	227	0.02	0.02	0.02	0.11	0.11	0.11	0.45	0.44	0.44	0.04

Table 6.8: Full comparison of results for the prediction task on 4 time series prediction datasets.

Data	Predictor	Symbols Length Number	Llama2-7B				Mistreal-7B				Informer				Time-LLM				TimeMixer
			r=16	r=64	r=256	r=16	r=64	r=256	r=16	r=64	r=256	[200]	[204]	[67]	[200]	[204]	[67]	[200]	[204]
			MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE
ETTh1	168/24	2,789	0.689 0.653	0.647 0.696	0.658 0.677	0.631 0.681	0.622 0.631	0.626 0.677	0.577 0.549	-	-	-	-	-	-	-	-	-	-
ETTh2	168/24	5,383	0.798 0.788	0.784 0.761	0.789 0.772	0.776 0.787	0.759 0.761	0.762 0.771	0.720 0.665	-	-	-	-	-	-	-	-	-	-
ETTm1	168/24	3,170	0.403 0.397	0.386 0.364	0.392 0.385	0.457 0.422	0.401 0.387	0.407 0.397	0.323 0.369	-	-	-	-	-	-	-	-	-	-
ETTm2	168/24	6,878	0.224 0.209	0.201 0.198	0.215 0.207	0.251 0.237	0.214 0.203	0.218 0.209	-	-	-	-	-	-	-	-	-	-	-
ETTh1	168/96	2,789	0.762 0.786	0.754 0.752	0.759 0.60	0.792 0.804	0.773 0.782	0.781 0.788	-	-	-	0.362	0.392	0.375	0.440	-	-	-	-
ETTh2	168/96	5,383	0.912 0.885	0.892 0.881	0.907 0.876	0.899 0.887	0.871 0.866	0.878 0.872	-	-	-	0.268	0.328	0.289	0.341	-	-	-	-
ETTm1	168/96	3,170	0.542 0.537	0.531 0.528	0.538 0.520	0.541 0.533	0.524 0.517	0.529 0.520	-	-	-	0.272	0.233	0.320	0.357	-	-	-	-
ETTm2	168/96	6,878	0.302 0.286	0.288 0.267	0.293 0.278	0.289 0.302	0.276 0.281	0.280 0.285	-	-	-	0.161	0.253	0.175	0.258	-	-	-	-
ETTh1	168/168	2,789	1.161 1.010	1.087 0.964	1.096 0.989	1.182 1.217	1.174 1.968	1.179 1.992	0.931 0.752	0.398	0.418	0.429	0.421	-	-	-	-	-	-
ETTh2	168/168	5,383	4.103 2.675	3.975 2.101	4.086 2.537	4.092 2.626	3.898 2.134	3.910 2.245	3.489 1.515	0.329	0.375	0.372	0.392	-	-	-	-	-	-
ETTm1	168/168	3,170	0.989 0.962	0.974 0.952	0.979 0.959	1.001 0.986	0.966 0.958	0.972 0.966	0.678 0.614	0.310	0.358	0.361	0.381	-	-	-	-	-	-
ETTm2	168/168	6,878	0.616 0.583	0.576 0.544	0.580 0.561	0.592 0.541	0.521 0.503	0.532 0.509	-	-	-	0.219	0.293	0.237	0.299	-	-	-	-

Table 6.9: The performance of LLM-ABBA with extra new tokens (symbolic ASCII codes) on ETTh1 data in terms of time series forecasting tasks.

Data	Predictor	Symbols Length Number	Mistreal-7B			
			r=16	r=64	r=256	
			MSE MAE	MSE MAE	MSE MAE	
ETTh1	168/24	2,789	0.636 0.692	0.626 0.632	0.629 0.681	
ETTh2	168/24	5,383	0.779 0.788	0.761 0.763	0.763 0.777	
ETTm1	168/24	3,170	0.457 0.402	0.402 0.387	0.407 0.399	
ETTm2	168/24	6,878	0.253 0.238	0.215 0.203	0.219 0.209	

Chapter 7

Conclusion

7.1 Summary

Depression is a complex, multisymptomatic, and highly recrudescient mental disease. Severity detection and psychotherapy have only been stated to be explored. In this work, we answered the big questions about scoring depressive severity and how to provide universal psychotherapy to depressive patients. In Chapter 2, we found that increased delta deactivation accompanied by strong beta activation is the main feature of depression as the severity of depression increases. We also verified that ANN models using EEGs can detect depression and score the severity of depression. In Chapter 3, we observed that the bilateral PFC plays a central role in various cognitive processes. For instance, it is involved in rehearsal activities prior to object recognition to aid classification. Additionally, the PFC facilitates inhibition to sustain positive memories and activities. It also supports disinhibition, which serves to stimulate or activate subsequent interactions within the brain. Meanwhile, the right PFC sometimes could assist left PFC to implement high capacity WM tasks. By contrast, the posterior regions, PPC, tends to be engaged in attention arousing and maintaining. These two findings suggest that **a)** the recurrent maintenance circuit may keep the brain performing positive cognitive components, **b)** then the instantaneous monitoring inhibition would pause the deadlocked sustenance function to save energy, and **c)** the arrival of disinhibition arouses the next step in the brain to select a new subject or focus on novel subjects. In Chapter 4, we addressed how pre-trained language models can enhance their performance on fine-tuning downstream tasks by modifying the attention block in Transformers. Additionally, we provided further evidence highlighting the importance of the inhibited gate mechanism in MLPs for effectively fine-tuning language downstream tasks. In Chapter 5, we observed that pretrained LLMs fine-tuned on Psychotherapy Assistant Instructions outperformed SOTA LLM response baselines. Our Assistant-Instruction approach introduces a semi-annotation method to

effectively align pretrained LLMs with instructional tasks. We also released a comprehensive synthetic dataset to support future research on professional instruction tuning tasks. In Chapter 6, we introduced ABBA-to-LLMs, a method that preserves the internal pattern structure of time series signals by transforming them into symbolic series that encapsulate the specific patterns of the original time series data. This method enables LLMs to interpret and analyze time series signals effectively. Finally, the methods and systems discussed form a closed-loop framework that integrates EEG signals with natural language processing.

7.2 Contributions and Achievements

The scientific contributions of this thesis are represented by the following achievements:

1. In Chapter 2, we investigated the neurophysiological differences between severe depression patients and healthy controls during working memory tasks, revealing a pronounced increase in central-parietal delta deactivation accompanied by strong beta activation in the depression group. Building on these findings, we proposed detection models based on specific EEG frequency bands and brain regions to classify depression and assess its severity. These models utilized scoring labels from two professional psychologists for validation, and the results were published in IEEE Transactions on Neural Systems and Rehabilitation Engineering. These findings were published on IEEE Transactions on Neural Systems and Rehabilitation Engineering (IF: 4.9, WOS citations: 3). The code and data of this project are available at <https://github.com/ChengKang520/Classifying-and-Scoring-MDD-BCI>.
2. In Chapter 3, we investigated WM brain networks using phase-lock coherence and directional coherence analyses. Adaptive fitting of 64-channel EEG data yielded four sources to simulate internal cerebral communications. Based on region-to-region connectivity, we proposed a “neurocognitive architecture” for WM, identifying pathways involved in memory maintenance and lateral inhibition. This methodology also demonstrates potential applications in depression detection and the visualization of abnormal brain activity. We published these findings on IEEE Transactions on Neural Systems and Rehabilitation Engineering (IF: 4.9, WOS citations: 15). The code and data of this project are available at <https://github.com/ChengKang520/Classifying-and-Scoring-MDD-BCI>.
3. In Chapter 4, we introduced a fine-tuning adaptation method, InA, designed to effectively suppress irrelevant information during fine-tuning on downstream tasks. InA

enhances the model’s focus on task-specific information by subtracting a threshold to eliminate the influence of extraneous knowledge. This approach is particularly applicable to fine-tuning models on professional psychotherapy data. We have published this method to Neural Networks (IF: 7.8, WOS citations: 1). The code and data of this project are available at <https://github.com/ChengKang520/inhibited-lora>.

4. In Chapter 5, we introduced psychotherapy data refined by GPT-4, enhancing LLMs’ comprehension of specialized professional knowledge and enabling them to generate content closely aligned with GPT-4 outputs. This chapter demonstrated the effectiveness of GPT-4-revised data for instruction-tuning LLMs, offering valuable insights for developing general-purpose, instruction-following agents powered by LLMs such as GPT-4. Additionally, this methodology is applicable to EHR tasks. The corresponding benchmark has been published at The 49th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) Workshop 2024 (Web of Science citations: 1). The code and data of this project are available at https://github.com/ChengKang520/psychotherapy-assistant_instruction/tree/main/Psych_BioGPT.
5. In Chapter 6, we introduced a time-series compression method designed to enhance the multimodal capabilities of LLMs for time-series analysis tasks. This tool improved LLMs’ ability to process time-series signals by leveraging the ABBA method for instruction-tuning. The results demonstrated that LLMs can effectively interpret the internal chain-of-patterns inherent in time-series data. Furthermore, this approach has potential applications in integrating EEG signals with LLMs. The proposed method has been submitted to IEEE Transactions on Signal Processing. The code and data of this project are available at <https://github.com/inEXASCALE/llm-abba>.

7.3 Future Work

In my future work, there are five main directions as follows.

- the development of a comprehensive toolbox capable of visualizing abnormal brain networks across various experimental paradigms;
- expanding the clinical data pool, optimizing models based on expert feedback, and enhancing the adaptability and deployability of large language models in psychotherapy, with a particular focus on depression intervention and adjunctive treatments;

- A valuable future direction is the practical application of the proposed psychotherapy chatbot to depression patients, enhancing the thesis's comprehensiveness by including real-world examples. This could involve showcasing how the chatbot supports depression management, aligning with the initial focus of the thesis. In addition, evaluating the performance of the chatbot would be essential, including comparative studies against existing tools to assess its efficacy in treating depression and providing therapeutic support;
- to enhance the performance of the psychotherapy-aiding chatbot across additional domains, such as auxiliary diagnosis, treatment recommendation support, and emotion monitoring through diary analysis.

List of Candidate's Publications Related to the Thesis

7.4 Publications in Impacted Journals

This thesis builds on the results previously published in the following publications:

1. **Kang, C.**; Novak, D.; Yao, X.; Xie, J.; Hu, Y#. (2023). "Classifying and Scoring Major Depressive Disorders by Residual Neural Networks on Specific Frequencies and Brain Regions," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 31, pp. 2964-2973, 2023, doi: 10.1109/TNSRE.2023.3293051.
2. **Kang, C.***; Li, Y.*; Novak, D.; Zhang, Y.; Zhou, Q.; Hu, Y#. (2020). "Brain Networks of Maintenance, Inhibition and Disinhibition During Working Memory," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 28, no. 7, pp. 1518-1527, 2020, doi: 10.1109/TNSRE.2020.2997827.
3. **Kang, C.#**; Prokop, J.; Tong, L.; Zhou, H.; Hu, Y.; Novak, D. (2024). InA: Inhibition Adaption on Pre-trained Language Models. Neural Networks, 178, 106410. <https://doi.org/10.1016/j.neunet.2024.106410>
4. **Kang, C.#**; Novak, D.; Urbanova, K.; Cheng, Y.; Hu, Y. (2024). Domain-Specific Improvement on Psychotherapy Chatbots Using Assistant. 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops, Seoul, Republic of Korea, 2024, pp. 351-355, doi: 10.1109/ICASSPW62465.2024.10626529.
5. Carson, E.; Cheng, X.#; **Kang, C#**. (2024). LLM-ABBA: Large Language Models Understand Time Series Via Symbolic Approximation. (Under review on IEEE Transactions on Signal Processing).

The following publications are related to the topic but were not included in the thesis, in order to keep the thesis more focused and easier to follow:

1. Yao, X.*; **Kang, C.***; Zhang, X.; Wang, S.; Zhang, Y#. (2024). FuzH-PID: Highly Controllable and Stable DNN for COVID-19 Detection via Improved Stochastic Optimization. *Expert Systems with Applications* (2024): 126323.
2. **Kang, C#**; Cheng, X.; Novak, D.; Yao, X. (2024, December). Using Laplace Transform to Optimize the Hallucination of Generation Models. In 2024 18th International Conference on Control, Automation, Robotics and Vision (ICARCV) (pp. 447-453). IEEE.
3. **Kang, C#.**; Yao, X.; Novak, D. (2023). Fuzzy Windows with Gaussian Process Labels for Ordinal Image Scoring Tasks. *Appl. Sci.* 2023, 13, 4019.
4. **Kang, C.**; Yu, X.; Wang, S. H.; Guttery, D.; Pandey, H.; Tian, Y.; Zhang, Y#. (2020). A heuristic neural network structure relying on fuzzy logic for images scoring. *IEEE Transactions on Fuzzy Systems.*, vol. 29, no. 1, pp. 34-45, Jan. 2021, doi: 10.1109/TFUZZ.2020.2966163.
5. Li, Y.*; **Kang, C.***; Wei, Z.; Qu, X.; Liu, T.; Zhou, Y.; Hu, Y#. (2017). Beta oscillations in major depression – signaling a new cortical circuit for central executive function. *Scientific reports*, 7 (1), 1-15, doi: 10.1038/s41598-017-18306-w.
6. Li, Y.*; **Kang, C.***; Qu, X.; Zhou, Y.; Wang, W.; Hu, Y#. (2016). Depression-related brain connectivity analyzed by EEG event-related phase synchrony measure. *Frontiers in human neuroscience*, 10, 477, doi: 10.3389/fnhum.2016.00477.

7.5 Other Publications

The following publications were published during the duration of the Ph.D. but are not included in the thesis because they are not directly related to the topic of the thesis:

1. Cui, H.; Li, H.; Li, G.; **Kang, C.**; Yao, X.; Feng, S.; Hu, Y#. (2019). Utility of trial-to-trial latency variability of somatosensory evoked potentials for diagnosis of spinal cord demyelination. *Journal of neurotrauma*, 36(24), 3356-3362, doi: 10.1089/neu.2018.6293.
2. Yu, X.; **Kang, C.**; Guttery, DS; Kadry, S.; Chen, Y.; Zhang, Y#. (2020). ResNet-SCDA-50 for breast abnormality classification. *IEEE / ACM transactions on computational biology and bioinformatics*, vol. 18, no. 1, pp. 94-102, 1 Jan.-Feb. 2021, doi: 10.1109/TCBB.2020.2986544.

3. Yao, X.; Zhu, Z.; **Kang, C.**; Wang, S.; Gorriz, J.; Zhang, Y#. (2022). AdaD-FNN for Chest CT-Based COVID-19 Diagnosis. IEEE Transactions on Emerging Topics in Computational Intelligence, doi: 10.1109/TETCI.2022.3174868.

The following publications were not included as they are currently under review:

1. **Kang, C.***; Yao, X.* (2023). Based on What We Can Control Artificial Neural Networks.
2. **Kang, C.#**; Chen, X.; Hu, Y.; Novak, D. (2024). Quantized Embedding Vectors for Controllable Diffusion Language Models.
3. Carson, E.; Chen, X.#; **Kang, C#**. (2024). Quantized symbolic time series approximation.

Bibliography

- [1] C. Kang, D. Novák, X. Yao, J. Xie, and Y. Hu, “Classifying and scoring major depressive disorders by residual neural networks on specific frequencies and brain regions”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 2964–2973, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:259369369>.
- [2] C. Kang, Y. Li, D. Novak, Y. Zhang, Q. Zhou, and Y. Hu, “Brain networks of maintenance, inhibition and disinhibition during working memory”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 7, pp. 1518–1527, 2020. DOI: [10.1109/TNSRE.2020.2997827](https://doi.org/10.1109/TNSRE.2020.2997827).
- [3] C. Kang, J. Prokop, L. Tong, H. Zhou, Y. Hu, and D. Novak, “Ina: Inhibition adaption on pre-trained language models”, *Neural Networks*, vol. 178, p. 106 410, 2024, ISSN: 0893-6080. DOI: <https://doi.org/10.1016/j.neunet.2024.106410>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608024003344>.
- [4] C. Kang, D. Novak, K. Urbanova, Y. Cheng, and Y. Hu, “Domain-specific improvement on psychotherapy chatbot using assistant”, in *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, 2024, pp. 351–355. DOI: [10.1109/ICASSPW62465.2024.10626529](https://doi.org/10.1109/ICASSPW62465.2024.10626529).
- [5] Y. Li, C. Kang, X. Qu, Y. Zhou, W. Wang, and Y. Hu, “Depression-related brain connectivity analyzed by eeg event-related phase synchrony measure”, *Frontiers in human neuroscience*, vol. 10, p. 477, 2016. DOI: <https://doi.org/10.3389/fnhum.2016.00477>.
- [6] Y. Li, C. Kang, Z. Wei, *et al.*, “Beta oscillations in major depression—signalling a new cortical circuit for central executive function”, *Scientific reports*, vol. 7, no. 1, pp. 1–15, 2017. DOI: <https://doi.org/10.1038/s41598-017-18306-w>.
- [7] C. Kang, J. Prokop, L. Tong, H. Zhou, Y. Hu, and D. Novak, “Gimlps: Gate with inhibition mechanism in mlps”, *arXiv preprint arXiv:2208.00929*, 2022.

- [8] C. Kang, X. Yu, S.-H. Wang, *et al.*, “A heuristic neural network structure relying on fuzzy logic for images scoring”, *IEEE transactions on fuzzy systems*, vol. 29, no. 1, pp. 34–45, 2020.
- [9] X. Yu, C. Kang, D. S. Guttery, S. Kadry, Y. Chen, and Y.-D. Zhang, “Resnet-scds-50 for breast abnormality classification”, *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 18, no. 1, pp. 94–102, 2020.
- [10] X. Yao, Z. Zhu, C. Kang, S.-H. Wang, J. M. Gorriz, and Y.-D. Zhang, “Adad-fnn for chest ct-based covid-19 diagnosis”, *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 1, pp. 5–14, 2022.
- [11] C. Kang, X. Yao, and D. Novak, “Fuzzy windows with gaussian processed labels for ordinal image scoring tasks”, *Applied Sciences*, vol. 13, no. 6, p. 4019, 2023.
- [12] X. Yao, C. Kang, X. Zhang, S. Wang, and Y. Zhang, “Fuzh-pid: Highly controllable and stable dnn for covid-19 detection via improved stochastic optimization”, *Expert Systems with Applications*, p. 126323, 2024.
- [13] E. Carson, X. Chen, and C. Kang, “Quantized symbolic time series approximation”, *arXiv preprint arXiv:2411.15209*, 2024.
- [14] E. Carson, X. Chen, and C. Kang, “Llm-abba: Understand time series via symbolic approximation”, *arXiv preprint arXiv:2411.18506*, 2024.
- [15] W. H. Organization. “Depressive disorder (depression)”. (2024), [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/depression> (visited on 08/30/2024).
- [16] S. Evans-Lacko, S. Aguilar-Gaxiola, A. Al-Hamzawi, *et al.*, “Socio-economic variations in the mental health treatment gap for people with anxiety, mood, and substance use disorders: Results from the who world mental health (wmh) surveys”, *Psychological medicine*, vol. 48, no. 9, pp. 1560–1571, 2018.
- [17] G. V. P. Reddy, “Depression – the global crisis”, vol. 34, no. 3, p. 201, 2013.
- [18] B. F. Grant, F. S. Stinson, D. A. Dawson, *et al.*, “Prevalence and co-occurrence of substance use disorders and independent mood and anxiety disorders: Results from the national epidemiologic survey on alcohol and related conditions”, vol. 29, no. 7, pp. 807–16, 2006.
- [19] M. Vermani, M. Marcus, and M. A. Katzman, “Rates of detection of mood and anxiety disorders in primary care: A descriptive, cross-sectional study”, vol. 13, no. 2, 2011.

- [20] J. C. Fournier, N. R. Forand, Z. Wang, *et al.*, “Initial severity and depressive relapse in cognitive behavioral therapy and antidepressant medications: An individual patient data meta-analysis”, *Cognitive Therapy and Research*, vol. 46, no. 3, pp. 517–531, 2022, ISSN: 1573-2819.
- [21] I. Kirsch, B. J. Deacon, T. B. Huedo-Medina, A. Scoboria, T. J. Moore, and B. T. Johnson, “Initial severity and antidepressant benefits: A meta-analysis of data submitted to the food and drug administration”, *PLoS medicine*, vol. 5, no. 2, e45, 2008, ISSN: 1549-1277.
- [22] H. S. Sharma, M. Chopp, L. Chen, *et al.*, “The 2021 yearbook of neurorestoratology”, *Journal of Neurorestoratology*, p. 100 008, 2022.
- [23] J. Davies and J. Read, “A systematic review into the incidence, severity and duration of antidepressant withdrawal effects: Are guidelines evidence-based?”, *Addictive behaviors*, vol. 97, pp. 111–121, 2019, ISSN: 0306-4603.
- [24] E. Hossain, R. Rana, N. Higgins, *et al.*, “Natural language processing in electronic health records in relation to healthcare decision-making: A systematic review”, *Computers in biology and medicine*, vol. 155, p. 106 649, 2023.
- [25] Y. Wu, M. Jiang, J. Xu, D. Zhi, and H. Xu, “Clinical named entity recognition using deep learning models”, in *AMIA annual symposium proceedings*, vol. 2017, 2018, p. 1812.
- [26] R. Pivovarov and N. Elhadad, “Automated methods for the summarization of electronic health records”, *Journal of the American Medical Informatics Association*, vol. 22, no. 5, pp. 938–947, 2015.
- [27] H. Alemzadeh and M. Devarakonda, “An nlp-based cognitive system for disease status identification in electronic health records”, in *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, IEEE, 2017, pp. 89–92.
- [28] C. Pandey, Z. Ibrahim, H. Wu, E. Iqbal, and R. Dobson, “Improving rnn with attention and embedding for adverse drug reactions”, in *Proceedings of the 2017 international conference on digital health*, 2017, pp. 67–71.
- [29] W. Shi, R. Xu, Y. Zhuang, *et al.*, “Ehrgent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records”, in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 22 315–22 339.

- [30] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, H. Adeli, and D. P. Subha, “Automated eeg-based screening of depression using deep convolutional neural network”, *Computer methods and programs in biomedicine*, vol. 161, pp. 103–113, 2018, ISSN: 0169-2607.
- [31] G. Andrews and L. Peters, “The psychometric properties of the composite international diagnostic interview”, vol. 33, no. 2, pp. 80–88,
- [32] M. B. First and M. Gibbon, “The structured clinical interview for dsm-iv axis i disorders (scid-i) and the structured clinical interview for dsm-iv axis ii disorders (scid-ii)”, 2004. [Online]. Available: <https://psycnet.apa.org/record/2004-12821-011>.
- [33] M. Zimmerman, J. H. Martinez, D. Young, I. Chelminski, and K. Dalrymple, “Severity classification on the hamilton depression rating scale”, *Journal of Affective Disorders*, vol. 150, no. 2, pp. 384–388, 2013, ISSN: 0165-0327. DOI: <https://doi.org/10.1016/j.jad.2013.04.028>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165032713003017>.
- [34] G. Jackson-Koku, “Beck depression inventory”, *Occupational Medicine*, vol. 66, no. 2, pp. 174–175, 2016.
- [35] M. Buoli, B. M. Cesana, J. L. Barkin, G. Tacchini, and A. C. Altamura, “Validity of a clinical diagnosis of bipolar disorder among participants in a multicenter study using the mini-international neuropsychiatric interview”, *Bipolar disorders*, vol. 20, no. 3, pp. 284–284, 2018, ISSN: 1398-5647.
- [36] B. K. Natamba, J. Achan, A. Arbach, *et al.*, “Reliability and validity of the center for epidemiologic studies-depression scale in screening for depression among hiv-infected and-uninfected pregnant women attending antenatal services in northern uganda: A cross-sectional study”, *BMC psychiatry*, vol. 14, no. 1, pp. 1–8, 2014, ISSN: 1471-244X.
- [37] S. Yasin, S. A. Hussain, S. Aslan, I. Raza, M. Muzammel, and A. Othmani, “Eeg based major depressive disorder and bipolar disorder detection using neural networks: A review”, *Computer Methods and Programs in Biomedicine*, vol. 202, p. 106 007, 2021.
- [38] M. Zimmerman, J. H. Martinez, D. Young, I. Chelminski, and K. Dalrymple, “Severity classification on the hamilton depression rating scale”, *Journal of affective disorders*, vol. 150, no. 2, pp. 384–388, 2013.
- [39] W. C. Drevets, J. L. Price, J. R. Simpson Jr, *et al.*, “Subgenual prefrontal cortex abnormalities in mood disorders”, *Nature*, vol. 386, no. 6627, pp. 824–827, 1997.

- [40] T. H. Ng, L. B. Alloy, and D. V. Smith, “Meta-analysis of reward processing in major depressive disorder reveals distinct abnormalities within the reward circuit”, *Translational psychiatry*, vol. 9, no. 1, p. 293, 2019.
- [41] J. P. Hamilton, A. Etkin, D. J. Furman, M. G. Lemus, R. F. Johnson, and I. H. Gotlib, “Functional neuroimaging of major depressive disorder: A meta-analysis and new integration of baseline activation and neural response data”, *American Journal of Psychiatry*, vol. 169, no. 7, pp. 693–703, 2012.
- [42] F. S. de Aguiar Neto and J. L. G. Rosa, “Depression biomarkers using non-invasive eeg: A review”, *Neuroscience & Biobehavioral Reviews*, vol. 105, pp. 83–93, 2019.
- [43] A.-C. Ehlis, S. Schneider, T. Dresler, and A. J. Fallgatter, “Application of functional near-infrared spectroscopy in psychiatry”, *Neuroimage*, vol. 85, pp. 478–488, 2014.
- [44] R. Wang, Y. Hao, Q. Yu, M. Chen, I. Humar, and G. Fortino, “Depression analysis and recognition based on functional near-infrared spectroscopy”, *IEEE journal of biomedical and health informatics*, vol. 25, no. 12, pp. 4289–4299, 2021.
- [45] C. M. Michel and D. Brunet, “Eeg source imaging: A practical review of the analysis steps”, *Frontiers in neurology*, vol. 10, p. 325, 2019.
- [46] N. Houlsby, A. Giurgiu, S. Jastrzebski, *et al.*, “Parameter-efficient transfer learning for nlp”, in *International Conference on Machine Learning*, PMLR, 2019, pp. 2790–2799.
- [47] X. L. Li and P. Liang, “Prefix-tuning: Optimizing continuous prompts for generation”, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. abs/2101.00190, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:230433941>.
- [48] E. J. Hu, Y. Shen, P. Wallis, *et al.*, “LoRA: Low-rank adaptation of large language models”, *arXiv:2106.09685*, 2021.
- [49] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, “Towards a unified view of parameter-efficient transfer learning”, *arXiv:2110.04366*, 2021.
- [50] B. Lester, R. Al-Rfou, and N. Constant, “The power of scale for parameter-efficient prompt tuning”, *arXiv:2104.08691*, 2021.
- [51] T. Brown, B. Mann, N. Ryder, *et al.*, “Language models are few-shot learners”, *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

- [52] J. Wei, X. Wang, D. Schuurmans, *et al.*, “Chain-of-thought prompting elicits reasoning in large language models”, *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.
- [53] M. Parmar, S. Mishra, M. Purohit, M. Luo, M. H. Murad, and C. Baral, “Inboxbart: Get instructions into biomedical multi-task learning”, *arXiv preprint arXiv:2204.07600*, 2022.
- [54] Y. Wang, S. Mishra, P. Alipoormolabashi, *et al.*, “Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks”, *arXiv:2204.07705*, 2022.
- [55] L. Ouyang, J. Wu, X. Jiang, *et al.*, “Training language models to follow instructions with human feedback”, *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.
- [56] Y. Wang, Y. Kordi, S. Mishra, *et al.*, “Self-instruct: Aligning language model with self generated instructions”, *arXiv preprint arXiv:2212.10560*, 2022.
- [57] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, “Large language models in medicine”, *Nature Medicine*, pp. 1–11, 2023.
- [58] L. Yang, H. Chen, Z. Li, X. Ding, and X. Wu, “Chatgpt is not enough: Enhancing large language models with knowledge graphs for fact-aware language modeling”, *arXiv preprint arXiv:2306.11489*, 2023.
- [59] R. Yang, T. F. Tan, W. Lu, A. J. Thirunavukarasu, D. S. W. Ting, and N. Liu, “Large language models in health care: Development, applications, and challenges”, *Health Care Science*, 2023.
- [60] O. Honovich, T. Scialom, O. Levy, and T. Schick, “Unnatural instructions: Tuning language models with (almost) no human labor”, *arXiv preprint arXiv:2212.09689*, 2022.
- [61] H. Touvron, T. Lavril, G. Izacard, *et al.*, “Llama: Open and efficient foundation language models”, *arXiv preprint arXiv:2302.13971*, 2023a.
- [62] H. Touvron, L. Martin, K. Stone, *et al.*, “Llama 2: Open foundation and fine-tuned chat models”, *arXiv preprint arXiv:2307.09288*, 2023b.
- [63] A. Dubey, A. Jauhri, A. Pandey, *et al.*, “The llama 3 herd of models”, *arXiv preprint arXiv:2407.21783*, 2024.
- [64] Y. Wang, S. Mishra, P. Alipoormolabashi, *et al.*, “Benchmarking generalization via in-context instructions on 1,600+ language tasks”, *arXiv e-prints*, arXiv–2204, 2022.

- [65] B. Peng, C. Li, P. He, M. Galley, and J. Gao, “Instruction tuning with gpt-4”, *arXiv preprint arXiv:2304.03277*, 2023.
- [66] M. Jin, S. Wang, L. Ma, *et al.*, “Time-LLM: Time series forecasting by reprogramming large language models”, in *The Twelfth International Conference on Learning Representations*, 2024.
- [67] S. Wang, H. Wu, X. Shi, *et al.*, “TimeMixer: Decomposable multiscale mixing for time series forecasting”, *arXiv preprint arXiv:2405.14616*, 2024.
- [68] T. Zhou, P. Niu, L. Sun, R. Jin, *et al.*, “One fits all: Power general time series analysis by pretrained lm”, *Advances in neural information processing systems*, vol. 36, pp. 43 322–43 355, 2023.
- [69] M. Jin, Y. Zhang, W. Chen, *et al.*, “Position: What can large language models tell us about time series analysis”, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:270213739>.
- [70] S. Malinowski, T. Guyet, R. Quiniou, and R. Tavenard, “1d-SAX: A novel symbolic representation for time series”, in *Advances in Intelligent Data Analysis XII*, 2013.
- [71] S. Elsworth and S. Güttel, “ABBA: adaptive Brownian bridge-based symbolic aggregation of time series”, *Data Mining and Knowledge Discovery*, vol. 34, pp. 1175–1200, 2020.
- [72] X. Chen and S. Güttel, “An efficient aggregation method for the symbolic representation of temporal data”, *ACM Transactions on Knowledge Discovery from Data*, 2022.
- [73] H. Mizuhara, L.-Q. Wang, K. Kobayashi, and Y. Yamaguchi, “Long-range eeg phase synchronization during an arithmetic task indexes a coherent cortical network simultaneously measured by fmri”, *Neuroimage*, vol. 27, no. 3, pp. 553–563, 2005, ISSN: 1053-8119.
- [74] H. Mizuhara and Y. Yamaguchi, “Human cortical circuits for central executive function emerge by theta phase synchronization”, *Neuroimage*, vol. 36, no. 1, pp. 232–244, 2007, ISSN: 1053-8119.
- [75] H. Cai, J. Han, Y. Chen, *et al.*, “A pervasive approach to eeg-based depression detection”, *Complexity*, vol. 2018, 2018, ISSN: 1076-2787.
- [76] H. Cai, Z. Qu, Z. Li, Y. Zhang, X. Hu, and B. Hu, “Feature-level fusion approaches based on multimodal eeg data for depression recognition”, *Information Fusion*, vol. 59, pp. 127–138, 2020, ISSN: 1566-2535.

- [77] S. D. Puthankattil and P. K. Joseph, “Classification of eeg signals in normal and depression conditions by ann using rwe and signal entropy”, *Journal of Mechanics in Medicine and biology*, vol. 12, no. 04, p. 1 240 019, 2012, ISSN: 0219-5194.
- [78] S. Aydın, “Cross-validated adaboost classification of emotion regulation strategies identified by spectral coherence in resting-state”, *Neuroinformatics*, pp. 1–13, 2021, ISSN: 1559-0089.
- [79] A. F. Leuchter, I. A. Cook, A. M. Hunter, C. Cai, and S. Horvath, “Resting-state quantitative electroencephalography reveals increased neurophysiologic connectivity in depression”, *PloS one*, vol. 7, no. 2, e32508, 2012, ISSN: 1932-6203.
- [80] B. Li, K. Friston, M. Mody, H. Wang, H. Lu, and D. Hu, “A brain network model for depression: From symptom understanding to disease intervention”, *CNS neuroscience & therapeutics*, vol. 24, no. 11, pp. 1004–1019, 2018, ISSN: 1755-5930.
- [81] P. Beloe and N. Derakshan, “Adaptive working memory training can reduce anxiety and depression vulnerability in adolescents”, *Developmental science*, e12831, 2019, ISSN: 1363-755X.
- [82] S. J. Bruijniks, G. van Grootheest, P. Cuijpers, *et al.*, “Working memory moderates the relation between the brain-derived neurotropic factor (bdnf) and psychotherapy outcome for depression”, *Journal of Psychiatric Research*, vol. 130, pp. 424–432, 2020, ISSN: 0022-3956.
- [83] K. Yoshida, Y. Shimizu, J. Yoshimoto, *et al.*, “Prediction of clinical depression scores and detection of changes in whole-brain using resting-state functional mri data with partial least squares regression”, *PloS one*, vol. 12, no. 7, e0179638, 2017, ISSN: 1932-6203.
- [84] M. Tanaka, Y. Shigihara, M. Funakura, E. Kanai, and Y. Watanabe, “Fatigue-associated alterations of cognitive function and electroencephalographic power densities”, *PLoS One*, vol. 7, no. 4, e34774, 2012, ISSN: 1932-6203.
- [85] A. Yassin, A.-H. Al-Mistarehi, K. El-Salem, *et al.*, “Clinical, radiological, and electroencephalographic features of hhv-6 encephalitis following hematopoietic stem cell transplantation”, *Annals of Medicine and Surgery*, vol. 60, pp. 81–86, 2020, ISSN: 2049-0801.
- [86] F. Zhang, F. Wang, C.-H. Li, *et al.*, “Therapeutic effects of subthalamic nucleus deep brain stimulation on anxiety and depression in parkinson’s disease patients”, *Journal of Neurorestoratology*, vol. 10, no. 1, pp. 31–42, 2022.

- [87] B. K. Prusty, N. Gulve, S. Govind, *et al.*, “Active hhv-6 infection of cerebellar purkinje cells in mood disorders”, *Frontiers in microbiology*, vol. 9, p. 1955, 2018, ISSN: 1664-302X.
- [88] N. Kobayashi, N. Oka, M. Takahashi, *et al.*, “Human herpesvirus 6b greatly increases risk of depression by activating hypothalamic-pituitary-adrenal axis during latent phase of infection”, *iScience*, p. 101 187, 2020, ISSN: 2589-0042.
- [89] O. Murphy, K. Hoy, D Wong, N. Bailey, P. B. Fitzgerald, and R. Segrave, “Individuals with depression display abnormal modulation of neural oscillatory activity during working memory encoding and maintenance”, *Biological psychology*, vol. 148, p. 107 766, 2019, ISSN: 0301-0511.
- [90] A. A. Fingelkurts and A. A. Fingelkurts, “Altered structure of dynamic electroencephalogram oscillatory pattern in major depression”, *Biological Psychiatry*, vol. 77, no. 12, pp. 1050–1060, 2015, ISSN: 0006-3223.
- [91] A. A. Fingelkurts, A. A. Fingelkurts, H. Rytsälä, K. Suominen, E. Isometsä, and S. Kähkönen, “Composition of brain oscillations in ongoing eeg during major depression disorder”, *Neuroscience research*, vol. 56, no. 2, pp. 133–144, 2006, ISSN: 0168-0102.
- [92] Y. Pathak, O. Salami, S. Baillet, Z. Li, and C. R. Butson, “Longitudinal changes in depressive circuitry in response to neuromodulation therapy”, *Frontiers in neural circuits*, vol. 10, p. 50, 2016, ISSN: 1662-5110.
- [93] K. Benchenane, P. H. Tiesinga, and F. P. Battaglia, “Oscillations in the prefrontal cortex: A gateway to memory and attention”, *Current opinion in neurobiology*, vol. 21, no. 3, pp. 475–485, 2011, ISSN: 0959-4388.
- [94] X. Zhang, J. Li, K. Hou, B. Hu, J. Shen, and J. Pan, “Eeg-based depression detection using convolutional neural network with demographic attention mechanism”, in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, pp. 128–133, ISBN: 1728119901.
- [95] X. Li, R. La, Y. Wang, *et al.*, “Eeg-based mild depression recognition using convolutional neural network”, *Medical & biological engineering & computing*, vol. 57, no. 6, pp. 1341–1352, 2019, ISSN: 1741-0444.
- [96] R. L. Spitzer, *Structured Clinical Interview for DSM-IV Axis I Disorders (SCID-I), Clinician Version, User’s Guide*. 1996.
- [97] B. Mwangi, K. Matthews, and J. D. Steele, “Prediction of illness severity in patients with major depression using structural mr brain scans”, *Journal of Magnetic Resonance Imaging*, vol. 35, no. 1, pp. 64–71, 2012, ISSN: 1053-1807.

- [98] C. Constantinidis and T. Klingberg, “The neuroscience of working memory capacity and training”, *Nature Reviews Neuroscience*, vol. 17, no. 7, pp. 438–449, 2016.
- [99] M. J. Kane, L. H. Brown, J. C. McVay, P. J. Silvia, I. Myin-Germeys, and T. R. Kwapil, “For whom the mind wanders, and when: An experience-sampling study of working memory and executive control in daily life”, *Psychological science*, vol. 18, no. 7, pp. 614–621, 2007.
- [100] S. E. Gathercole, L. Brown, and S. J. Pickering, “Working memory assessments at school entry as longitudinal predictors of national curriculum attainment levels”, *Educational and Child Psychology*, vol. 20, no. 3, pp. 109–122, 2003.
- [101] J. Eriksson, E. K. Vogel, A. Lansner, F. Bergström, and L. Nyberg, “Neurocognitive architecture of working memory”, *Neuron*, vol. 88, no. 1, pp. 33–46, 2015.
- [102] M. D’Esposito and B. R. Postle, “The cognitive neuroscience of working memory”, *Annual review of psychology*, vol. 66, pp. 115–142, 2015.
- [103] T. Pasternak and M. W. Greenlee, “Working memory in primate sensory systems”, *Nature Reviews Neuroscience*, vol. 6, no. 2, pp. 97–107, 2005.
- [104] R. A. Charlton, T. R. Barrick, I. N. C. Lawes, H. S. Markus, and R. G. Morris, “White matter pathways associated with working memory in normal aging”, *Cortex*, vol. 46, no. 4, pp. 474–489, 2010.
- [105] D. E. Nee, J. W. Brown, M. K. Askren, *et al.*, “A meta-analysis of executive components of working memory”, *Cerebral cortex*, vol. 23, no. 2, pp. 264–282, 2013.
- [106] N. Dolu, C Başar-Eroğlu, Ç Özesmi, and C Süer, “An assessment of working memory using p300 wave in healthy subjects”, in *International Congress Series*, Elsevier, vol. 1278, 2005, pp. 7–10.
- [107] A. M. Owen, K. M. McMillan, A. R. Laird, and E. Bullmore, “N-back working memory paradigm: A meta-analysis of normative functional neuroimaging studies”, *Human brain mapping*, vol. 25, no. 1, pp. 46–59, 2005.
- [108] T. D. Wager and E. E. Smith, “Neuroimaging studies of working memory”, *Cognitive, Affective, & Behavioral Neuroscience*, vol. 3, pp. 255–274, 2003.
- [109] K. Kubota and H. Niki, “Prefrontal cortical unit activity and delayed alternation performance in monkeys.”, *Journal of neurophysiology*, vol. 34, no. 3, pp. 337–347, 1971.
- [110] J. M. Fuster and G. E. Alexander, “Neuron activity related to short-term memory”, *Science*, vol. 173, no. 3997, pp. 652–654, 1971.

- [111] U. Leon-Dominguez, J. F. Martín-Rodríguez, and J. León-Carrión, “Executive n-back tasks for the neuropsychological assessment of working memory”, *Behavioural brain research*, vol. 292, pp. 167–173, 2015.
- [112] F. Collette, M. Hogge, E. Salmon, and M. Van der Linden, “Exploration of the neural substrates of executive functioning by functional neuroimaging”, *Neuroscience*, vol. 139, no. 1, pp. 209–221, 2006.
- [113] F. Collette, M. Van der Linden, S. Laureys, *et al.*, “Exploring the unity and diversity of the neural substrates of executive functioning”, *Human brain mapping*, vol. 25, no. 4, pp. 409–423, 2005.
- [114] M. Koenigs, A. K. Barbey, B. R. Postle, and J. Grafman, “Superior parietal cortex is critical for the manipulation of information in working memory”, *Journal of Neuroscience*, vol. 29, no. 47, pp. 14 980–14 986, 2009.
- [115] B. R. Buchsbaum and M. D’Esposito, “The search for the phonological store: From loop to convolution”, *Journal of Cognitive Neuroscience*, vol. 20, no. 5, pp. 762–778, 2008.
- [116] N. E. Myers, M. G. Stokes, and A. C. Nobre, “Prioritizing information during working memory: Beyond sustained internal attention”, *Trends in cognitive sciences*, vol. 21, no. 6, pp. 449–461, 2017.
- [117] A. Ikkai and C. E. Curtis, “Common neural mechanisms supporting spatial working memory, attention and motor intention”, *Neuropsychologia*, vol. 49, no. 6, pp. 1428–1434, 2011.
- [118] T. A. Jerde, E. P. Merriam, A. C. Riggall, J. H. Hedges, and C. E. Curtis, “Prioritized maps of space in human frontoparietal cortex”, *Journal of Neuroscience*, vol. 32, no. 48, pp. 17 382–17 390, 2012.
- [119] J. A. Cromer, J. E. Roy, T. J. Buschman, and E. K. Miller, “Comparison of primate prefrontal and premotor cortex neuronal activity during visual categorization”, *Journal of cognitive neuroscience*, vol. 23, no. 11, pp. 3355–3365, 2011.
- [120] J. E. Roy, M. Riesenhuber, T. Poggio, and E. K. Miller, “Prefrontal cortex activity during flexible categorization”, *Journal of Neuroscience*, vol. 30, no. 25, pp. 8519–8528, 2010.
- [121] M. G. Stokes, M. Kusunoki, N. Sigala, H. Nili, D. Gaffan, and J. Duncan, “Dynamic coding for cognitive control in prefrontal cortex”, *Neuron*, vol. 78, no. 2, pp. 364–375, 2013.

- [122] R. Quentin, J.-R. King, E. Sallard, *et al.*, “Differential brain mechanisms of selection and maintenance of information during working memory”, *Journal of Neuroscience*, vol. 39, no. 19, pp. 3728–3740, 2019.
- [123] J. M. Fuster and S. L. Bressler, “Cognit activation: A mechanism enabling temporal integration in working memory”, *Trends in cognitive sciences*, vol. 16, no. 4, pp. 207–218, 2012.
- [124] G. S. Shields, J. C. Bonner, and W. G. Moons, “Does cortisol influence core executive functions? a meta-analysis of acute cortisol administration effects on working memory, inhibition, and set-shifting”, *Psychoneuroendocrinology*, vol. 58, pp. 91–103, 2015.
- [125] C. Rottschy, R. Langner, I. Dogan, *et al.*, “Modelling neural correlates of working memory: A coordinate-based meta-analysis”, *Neuroimage*, vol. 60, no. 1, pp. 830–846, 2012.
- [126] L. Hu, Z. Zhang, and Y. Hu, “A time-varying source connectivity approach to reveal human somatosensory information processing”, *Neuroimage*, vol. 62, no. 1, pp. 217–228, 2012.
- [127] F. Li, B. Chen, H. Li, *et al.*, “The time-varying networks in p300: A task-evoked eeg study”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 7, pp. 725–733, 2016.
- [128] G Tropini, J Chiang, Z. Wang, and M. McKeown, “Partial directed coherence-based information flow in parkinson’s disease patients performing a visually-guided motor task”, in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2009, pp. 1873–1878.
- [129] H. Mizuhara and Y. Yamaguchi, “Human cortical circuits for central executive function emerge by theta phase synchronization”, *NeuroImage*, vol. 36, no. 1, pp. 232–244, 2007, ISSN: 1053-8119. DOI: <https://doi.org/10.1016/j.neuroimage.2007.02.026>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1053811907001085>.
- [130] L. A. Baccalá and K. Sameshima, “Partial directed coherence: A new concept in neural structure determination”, *Biological cybernetics*, vol. 84, no. 6, pp. 463–474, 2001.
- [131] L. A. Baccala, K. Sameshima, and D. Y. Takahashi, “Generalized partial directed coherence”, in *2007 15th International conference on digital signal processing*, Ieee, 2007, pp. 163–166.

- [132] J. Dauwels, F. B. Vialatte, T. Musha, and A. Cichocki, “A comparative study of synchrony measures for the early diagnosis of alzheimer’s disease based on eeg”, *NeuroImage*, vol. 49, pp. 668–693, 2010. [Online]. Available: <https://api.semanticscholar.org/CorpusID:5708896>.
- [133] B. Scott L and M. Earl K, “Frequency-specific hippocampalprefrontal interactions during associative learning”, vol. 18, Springer, 2015, pp. 576–581.
- [134] P.-O. Harvey, P. Fossati, J.-B. Pochon, *et al.*, “Cognitive control and brain resources in major depression: An fmri study using the n-back task”, *NeuroImage*, vol. 26, pp. 860–869, 2005. [Online]. Available: <https://api.semanticscholar.org/CorpusID:39617212>.
- [135] M. P. Tarvainen, J. K. Hiltunen, P. O. Ranta-aho, and P. A. Karjalainen, “Estimation of nonstationary eeg with kalman smoother approach: An application to event-related synchronization (ers)”, *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 516–524, 2004. [Online]. Available: <https://api.semanticscholar.org/CorpusID:5423390>.
- [136] J. Polich, “Updating p300: An integrative theory of p3a and p3b”, *Clinical Neurophysiology*, vol. 118, pp. 2128–2148, 2007. [Online]. Available: <https://api.semanticscholar.org/CorpusID:9753972>.
- [137] C. J. Stoodley and J. D. Schmahmann, “Functional topography in the human cerebellum: A meta-analysis of neuroimaging studies”, *NeuroImage*, vol. 44, pp. 489–501, 2009. [Online]. Available: <https://api.semanticscholar.org/CorpusID:2703288>.
- [138] J. J. LaRocque, J. A. Lewis-Peacock, A. T. Drysdale, K. Oberauer, and B. R. Postle, “Decoding attended information in short-term memory: An eeg study”, *Journal of Cognitive Neuroscience*, vol. 25, pp. 127–142, 2013. [Online]. Available: <https://api.semanticscholar.org/CorpusID:1071449>.
- [139] O. Barak and M. Tsodyks, “Working models of working memory”, *Current Opinion in Neurobiology*, vol. 25, pp. 20–24, 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:31958359>.
- [140] R. C. O’Reilly, “Biologically based computational models of high-level cognition”, *Science*, vol. 314, pp. 91 –94, 2006. [Online]. Available: <https://api.semanticscholar.org/CorpusID:11620257>.
- [141] A. K. Engel and P. Fries, “Beta-band oscillations—signalling the status quo?”, *Current Opinion in Neurobiology*, vol. 20, pp. 156–165, 2010. [Online]. Available: <https://api.semanticscholar.org/CorpusID:79336156>.

- [142] M. M. Shafi, Y. Zhou, J. Quintana, C. C. Chow, J. M. Fuster, and M. Bodner, “Variability in neuronal activity in primate cortex during working memory tasks”, *Neuroscience*, vol. 146, pp. 1082–1108, 2007. [Online]. Available: <https://api.semanticscholar.org/CorpusID:16256426>.
- [143] A. C. Riggall and B. R. Postle, “The relationship between working memory storage and elevated activity as measured with functional magnetic resonance imaging”, *The Journal of Neuroscience*, vol. 32, pp. 12990–12998, 2012. [Online]. Available: <https://api.semanticscholar.org/CorpusID:18395926>.
- [144] B. H. Silverstein, M. D. Snodgrass, H. Shevrin, and R. K. Kushwaha, “P3b, consciousness, and complex unconscious processing”, *Cortex*, vol. 73, pp. 216–227, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:206985968>.
- [145] L. Nyberg, M. Andersson, K. Kauppi, *et al.*, “Age-related and genetic modulation of frontal cortex efficiency”, *Journal of Cognitive Neuroscience*, vol. 26, pp. 746–754, 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:13583069>.
- [146] A. Compte, N. J.-B. Brunel, P. S. Goldman-Rakic, and X.-J. Wang, “Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model.”, *Cerebral cortex*, vol. 10 9, pp. 910–23, 2000. [Online]. Available: <https://api.semanticscholar.org/CorpusID:7239548>.
- [147] M. Wang, Y. Yang, C.-J. Wang, *et al.*, “Nmda receptors subserve persistent neuronal firing during working memory in dorsolateral prefrontal cortex”, *Neuron*, vol. 77, pp. 736–749, 2013. [Online]. Available: <https://api.semanticscholar.org/CorpusID:5659560>.
- [148] J. D. Murray, A. Anticevic, M. Gancsos, *et al.*, “Linking microcircuit dysfunction to cognitive impairment: Effects of disinhibition associated with schizophrenia in a cortical working memory model.”, *Cerebral cortex*, vol. 24 4, pp. 859–72, 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:2630534>.
- [149] M. Starc, J. D. Murray, N. Santamauro, *et al.*, “Schizophrenia is associated with a pattern of spatial working memory deficits consistent with cortical disinhibition”, *Schizophrenia Research*, vol. 181, pp. 107–116, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:3728957>.
- [150] P. M. Bays, “Spikes not slots: Noise in neural populations limits working memory”, *Trends in Cognitive Sciences*, vol. 19, pp. 431–438, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:698568>.

- [151] J. L. Lobo, J. Del Ser, A. Bifet, and N. Kasabov, “Spiking neural networks and online learning: An overview and perspectives”, *Neural Networks*, vol. 121, pp. 88–100, 2020.
- [152] L. J. Borg-Graham, C. Monier, and Y. Fregnac, “Visual input evokes transient and strong shunting inhibition in visual cortical neurons”, *Nature*, vol. 393, no. 6683, pp. 369–373, 1998.
- [153] W. Huang, Y. Ke, J. Zhu, *et al.*, “Tresk channel contributes to depolarization-induced shunting inhibition and modulates epileptic seizures”, *Cell Reports*, vol. 36, no. 3, p. 109404, 2021.
- [154] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need”, *Advances in neural information processing systems*, vol. 30, 2017.
- [155] J. Kaplan, S. McCandlish, T. Henighan, *et al.*, “Scaling laws for neural language models”, *arXiv:2001.08361*, 2020.
- [156] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding”, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, 2019, pp. 4171–4186.
- [157] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, “Language models are unsupervised multitask learners”, *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [158] C. Raffel, N. Shazeer, A. Roberts, *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer”, *arXiv:1910.10683*, 2019.
- [159] S. Smith, M. Patwary, B. Norick, *et al.*, “Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model”, *arXiv:2201.11990*, 2022.
- [160] Y. Liu, M. Ott, N. Goyal, *et al.*, “Roberta: A robustly optimized bert pretraining approach”, *arXiv:1907.11692*, 2019.
- [161] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “Albert: A lite bert for self-supervised learning of language representations”, *arXiv preprint arXiv:1909.11942*, 2019.
- [162] P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-enhanced BERT with disentangled attention”, *arXiv:2006.03654*, 2020.
- [163] P. He, J. Gao, and W. Chen, “DeBERTaV3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing”, *arXiv:2111.09543*, 2021.

- [164] W. Fedus, B. Zoph, and N. Shazeer, “Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity”, *arXiv:2101.03961*, 2021.
- [165] Z. Liu, Y. Lin, Y. Cao, *et al.*, “Swin transformer: Hierarchical vision transformer using shifted windows”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.
- [166] Z. Liu, H. Hu, Y. Lin, *et al.*, “Swin transformer V2: Scaling up capacity and resolution”, *arXiv:2111.09883*, 2021.
- [167] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, “Adapterfusion: Non-destructive task composition for transfer learning”, *arXiv:2005.00247*, 2020.
- [168] E. B. Zaken, S. Ravfogel, and Y. Goldberg, “BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models”, *arXiv:2106.10199*, 2021.
- [169] D. Guo, A. M. Rush, and Y. Kim, “Parameter-efficient transfer learning with diff pruning”, *arXiv:2012.07463*, 2020.
- [170] A. Chavan, Z. Liu, D. Gupta, E. Xing, and Z. Shen, “One-for-all: Generalized lora for parameter-efficient fine-tuning”, *arXiv preprint arXiv:2306.07967*, 2023.
- [171] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “Qlora: Efficient fine-tuning of quantized llms”, *arXiv preprint arXiv:2305.14314*, 2023.
- [172] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, “Going deeper in spiking neural networks: Vgg and residual architectures”, *Frontiers in neuroscience*, vol. 13, p. 95, 2019.
- [173] N. Rathi and K. Roy, “Diet-snn: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization”, *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [174] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, “Enabling spike-based backpropagation for training deep neural network architectures”, *Frontiers in neuroscience*, p. 119, 2020.
- [175] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, “Conversion of continuous-valued deep networks to efficient event-driven networks for image classification”, *Frontiers in neuroscience*, vol. 11, p. 682, 2017.
- [176] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, “Incorporating learnable membrane time constant to enhance learning of spiking neural networks”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2661–2671.

- [177] T. Wolf, L. Debut, V. Sanh, *et al.*, “Transformers: State-of-the-art natural language processing”, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online: Association for Computational Linguistics, 2020, pp. 38–45.
- [178] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “GLUE: a multi-task benchmark and analysis platform for natural language understanding”, *arXiv:1804.07461*, 2018.
- [179] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text”, *arXiv:1606.05250*, 2016.
- [180] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, “SWAG: A large-scale adversarial dataset for grounded commonsense inference”, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2018, pp. 93–104.
- [181] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, “Spanbert: Improving pre-training by representing and predicting spans”, *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.
- [182] D. P. Kingma, “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.
- [183] P. Rajpurkar, R. Jia, and P. Liang, “Know what you don’t know: Unanswerable questions for squad”, *arXiv:1806.03822*, 2018.
- [184] H. Liu, Z. Dai, D. So, and Q. V. Le, “Pay attention to mlps”, in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 9204–9215.
- [185] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units”, *arXiv:1508.07909*, 2015.
- [186] W. Qi, H. Fan, H. R. Karimi, and H. Su, “An adaptive reinforcement learning-based multimodal data fusion framework for human–robot confrontation gaming”, *Neural Networks*, vol. 164, pp. 489–496, 2023.
- [187] M. R. Pacheco-Lorenzo, S. M. Valladares-Rodríguez, L. E. Anido-Rifón, and M. J. Fernández-Iglesias, “Smart conversational agents for the detection of neuropsychiatric disorders: A systematic review”, *Journal of Biomedical Informatics*, vol. 113, p. 103632, 2021.
- [188] K. T. Pham, A. Nabizadeh, and S. Selek, “Artificial intelligence and chatbots in psychiatry”, *Psychiatric Quarterly*, vol. 93, no. 1, pp. 249–253, 2022.

- [189] A. Mallol-Ragolta, Z. Zhao, L. Stappen, N. Cummins, and B. Schuller, “A hierarchical attention network-based approach for depression detection from transcribed clinical interviews”, 2019.
- [190] Y.-T. Tsai and W.-A. Lin, “Design of an intelligent cognition assistant for people with cognitive impairment”, in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, IEEE, 2018, pp. 1207–1212.
- [191] A. N. Vaidyam, H. Wisniewski, J. D. Halamka, M. S. Kashavan, and J. B. Torous, “Chatbots and conversational agents in mental health: A review of the psychiatric landscape”, *The Canadian Journal of Psychiatry*, vol. 64, no. 7, pp. 456–464, 2019.
- [192] A. Das, S. Selek, A. R. Warner, *et al.*, “Conversational bots for psychotherapy: A study of generative transformer models using domain-specific dialogues”, in *Proceedings of the 21st Workshop on Biomedical Language Processing*, 2022, pp. 285–297.
- [193] O. Honovich, U. Shaham, S. R. Bowman, and O. Levy, “Instruction induction: From few examples to natural language task descriptions”, *arXiv preprint arXiv:2205.10782*, 2022.
- [194] S. Ye, D. Kim, J. Jang, J. Shin, and M. Seo, “Guess the instruction! flipped learning makes language models stronger zero-shot learners”, in *The Eleventh International Conference on Learning Representations*, 2022.
- [195] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-rag: Learning to retrieve, generate, and critique through self-reflection”, *arXiv preprint arXiv:2310.11511*, 2023.
- [196] Z. Du, Y. Qian, X. Liu, *et al.*, “Glm: General language model pretraining with autoregressive blank infilling”, in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 320–335.
- [197] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries”, in *Text summarization branches out*, 2004, pp. 74–81.
- [198] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto, “Diffusion-LM improves controllable text generation”, *arXiv:2205.14217*, 2022.
- [199] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, “Deep learning for time series classification: A review”, *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

- [200] C. W. Tan, C. Bergmeir, F. Petitjean, and G. I. Webb, “Time series extrinsic regression: Predicting numeric values from time series data”, *Data Mining and Knowledge Discovery*, vol. 35, no. 3, pp. 1032–1060, 2021.
- [201] A. A. Ismail, M. Gunady, H. Corrada Bravo, and S. Feizi, “Benchmarking deep learning interpretability in time series predictions”, *Advances in neural information processing systems*, vol. 33, pp. 6441–6452, 2020.
- [202] M. Jin, Y. Zhang, W. Chen, *et al.*, *Position paper: What can large language models tell us about time series analysis*, 2024. arXiv: [2402.02713 \[cs.LG\]](#).
- [203] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, “A time series is worth 64 words: Long-term forecasting with transformers”, *arXiv preprint arXiv:2211.14730*, 2022.
- [204] M. Jin, S. Wang, L. Ma, *et al.*, “Time-LLM: Time series forecasting by reprogramming large language models”, *arXiv preprint arXiv:2310.01728*, 2023.
- [205] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, “Large language models are zero-shot time series forecasters”, *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [206] K. Rasul, A. Ashok, A. R. Williams, *et al.*, “Lag-llama: Towards foundation models for time series forecasting”, *arXiv preprint arXiv:2310.08278*, 2023.
- [207] V. Ekambaram, A. Jati, P. Dayama, *et al.*, *Tiny Time Mixers (TTMs): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series*, 2024. arXiv: [2401.03955 \[cs.LG\]](#).
- [208] S. Mirchandani, F. Xia, P. Florence, *et al.*, “Large language models as general pattern machines”, *arXiv preprint arXiv:2307.04721*, 2023.
- [209] D. Spathis and F. Kawsar, “The first step is the hardest: Pitfalls of representing and tokenizing temporal data for large language models”, *Journal of the American Medical Informatics Association*, vol. 31, no. 9, pp. 2151–2158, 2024.
- [210] J. Lin, E. Keogh, L. Wei, and S. Lonardi, “Experiencing SAX: A novel symbolic representation of time series”, *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, 2007.
- [211] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “QLoRA: Efficient Finetuning of Quantized LLMs”, *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [212] Y. Liu, G. Qin, X. Huang, J. Wang, and M. Long, “AutoTimes: Autoregressive time series forecasters via large language models”, *arXiv preprint arXiv:2402.02370*, 2024.

- [213] X. Liu, J. Hu, Y. Li, *et al.*, “Unitime: A language-empowered unified model for cross-domain time series forecasting”, in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 4095–4106.
- [214] H. Xue and F. D. Salim, “PromptCast: A new prompt-based learning paradigm for time series forecasting”, *IEEE Transactions on Knowledge and Data Engineering*, 2023.
- [215] D. Cao, F. Jia, S. O. Arik, *et al.*, “Tempo: Prompt-based generative pre-trained transformer for time series forecasting”, *arXiv preprint arXiv:2310.04948*, 2023.
- [216] R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning, *et al.*, “STL: A seasonal-trend decomposition”, *Journal of Official Statistics*, vol. 6, no. 1, pp. 3–73, 1990.
- [217] A. van den Oord, S. Dieleman, H. Zen, *et al.*, “WaveNet: A Generative Model for Raw Audio”, in *Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, 2016, p. 125.
- [218] D. M. W. Powers, “Applications and explanations of Zipf’s law”, in *New Methods in Language Processing and Computational Natural Language Learning*, 1998.
- [219] C. Kang, J. Prokop, L. Tong, H. Zhou, Y. Hu, and D. Novak, “InA: Inhibition adaption on pre-trained language models”, *Neural Networks*, p. 106 410, 2024.
- [220] H. Touvron, T. Lavril, G. Izacard, *et al.*, “Llama: Open and efficient foundation language models”, *arXiv preprint arXiv:2302.13971*, 2023.
- [221] A. Q. Jiang, A. Sablayrolles, A. Mensch, *et al.*, “Mistral 7b”, *arXiv preprint arXiv:2310.06825*, 2023.
- [222] H. A. Dau, A. Bagnall, K. Kamgar, *et al.*, “The ucr time series archive”, *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, 2019.
- [223] A. Seyfi, J.-F. Rajotte, and R. Ng, “Generating multivariate time series with COMon Source Coordinated GAN (COSCI-GAN)”, *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 777–32 788, 2022.
- [224] S. Mousavi and F. Afghah, “Inter-and intra-patient ecg heartbeat classification for arrhythmia detection: A sequence to sequence deep learning approach”, in *IEEE international conference on acoustics, speech and signal processing*, IEEE, 2019, pp. 1308–1312.
- [225] Z. Liu and X. Zhang, “ECG-based heart arrhythmia diagnosis through attentional convolutional neural networks”, in *2021 IEEE International Conference on Internet of Things and Intelligence Systems (IoTIS)*, IEEE, 2021, pp. 156–162.

- [226] C.-H. H. Yang, Y.-Y. Tsai, and P.-Y. Chen, “Voice2series: Reprogramming acoustic models for time series classification”, in *International conference on machine learning*, PMLR, 2021, pp. 11 808–11 819.
- [227] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, “ECG heartbeat classification: A deep transferable representation”, *IEEE International Conference on Healthcare Informatics*, pp. 443–444, 2018.
- [228] S. P. Shashikumar, A. J. Shah, G. D. Clifford, and S. Nemati, “Detection of paroxysmal atrial fibrillation using attention-based bidirectional recurrent neural networks”, in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 715–723.
- [229] S. Singh, S. K. Pandey, U. Pawar, and R. R. Janghel, “Classification of ecg arrhythmia using recurrent neural networks”, *Procedia Computer Science*, vol. 132, pp. 1290–1297, 2018.
- [230] S. Saadatnejad, M. Oveisi, and M. Hashemi, “Lstm-based ecg classification for continuous monitoring on personal wearable devices”, *IEEE journal of biomedical and health informatics*, vol. 24, no. 2, pp. 515–523, 2019.
- [231] H. Zhou, S. Zhang, J. Peng, *et al.*, “Informer: Beyond efficient transformer for long sequence time-series forecasting”, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 11 106–11 115, 2021.
- [232] H. Wu, J. Xu, J. Wang, and M. Long, “Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting”, *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.
- [233] D. Jain, R. Ranjan, A. Sharma, S. N. Sharma, and A. Jain, “Fast and accurate ecg signal peaks detection using symbolic aggregate approximation”, *Multimedia Tools and Applications*, vol. 83, no. 30, pp. 75 033–75 059, 2024. DOI: [10.1007/s11042-024-18302-z](https://doi.org/10.1007/s11042-024-18302-z).
- [234] V. Jha and P. Tripathi, “Probabilistic sax: A cognitively-inspired method for time series classification in cognitive iot sensor network”, *Mobile Networks and Applications*, 2024. DOI: [10.1007/s11036-024-02322-y](https://doi.org/10.1007/s11036-024-02322-y).