

Finite Automata Models: Algorithm, Application, and Semigroup Study

by

Hanna Derets

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Applied Mathematics

Waterloo, Ontario, Canada, 2024

© Hanna Derets 2024

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Hanna Derets is the sole author of this thesis written under the supervision of Professor Chrystopher L. Nehaniv and none of the thesis was written for publication, except for parts of Chapter 4 as described below on this page.

This thesis includes first-authored peer-reviewed material accepted for publication in an edited conference proceedings volume, published by Springer Nature.

Research presented in Chapter 4 sections 4.3 and 4.4 (excluding 4.3.3):

Hanna Derets formulated the problems, carried out the research and drafted the manuscript. Chrystopher L. Nehaniv reviewed and edited the manuscript, and also supervised the work.

Citation:

H. Derets and C. L. Nehaniv, The study of the transformation semigroup of the abelian and directed non-abelian sandpiles. In *Recent Developments in Mathematical, Statistical and Computational Sciences (AMMCS 2023)*, D. M. Kilgour, H. Kunze, R. Makarov, R. Melnik, and X. Wang, Eds., Springer Proceedings in Mathematics & Statistics, Springer Nature, (in press).

Abstract

A finite automaton consists of states and state transitions labeled by letters of a finite alphabet. Every letter describes a transformation of the automaton's state space, generating a corresponding transformation semigroup. A stochastic version of automata has a probability distribution over the alphabet at every state, that allows assigning the likelihoods to generated sequences of letters. In this thesis, we explore two models: deterministic probabilistic finite automaton (DPFA) considered from the practical side, and the sandpile model considered from the theoretical side.

For the first model, the presented study describes an algorithm for reconstructing DPFA from sequences of discrete observations using an n -gram merging method, including the practical implementation of the method, its experimental evaluation on case studies, and the application of this automata-based technique to the neurobiological data. For considered examples the performance is compared to the causal state splitting reconstruction (CSSR) technique: both methods achieve high quality in approximating the probability distribution over strings. Considering if transformation semigroups of reconstructed automata contain subgroups corresponding to those in examples, CSSR shows a good result for preserving cyclic permutation groups, but not the n -gram merging method, whose transformation semigroup is generally aperiodic. The application considered in this study uses electroencephalographic (EEG) microstate sequences and considers the question of distinguishing the participant groups (meditators and controls) and cognitive modes (mind-wandering, verbalization, visualization) by separating DPFA machines inferred from EEG data in a metric space. The separation between participant groups is achieved for many parameter settings with linear criterion (requiring non-overlapping clusters) and for a few instances with strict criterion (requiring dense distant clusters). Both criteria show great reliability when validated using permuted data. The separation of cognitive modes only demonstrated partial success with noticeably better performance within the group of controls and more instances of separation corresponding to the visualization condition.

For the second, sandpile model, the presented study concentrates on the properties of their transformation semigroups for the standard Abelian sandpiles on circle graphs and the modified model, non-Abelian sandpiles on rooted trees. The exploration addresses the structure of recurrent configurations, the wreath product decomposition of semigroups, and the decomposition-based complexity measure. The identity and generator configurations of the recurrent group of Abelian sandpiles on circles are described, giving an explicit alternative way of understanding their well-known cyclic structure. The complexity of arbitrary finite Abelian semigroup is shown to be at most one. The embedding of the sandpile semigroup into the wreath product of flip-flop semigroups is constructed for non-Abelian sandpiles on rooted trees, implying its aperiodic complexity is the depth of the tree.

Acknowledgements

I am sincerely grateful to my supervisor, Professor Chrystopher L. Nehaniv, for his guidance, advice, and support throughout my studies. His thoughtful feedback and expertise were crucial in shaping the direction of my research and ensuring its quality.

I also wish to thank Dr. Elena Antonova for her advice and suggestions regarding the neurobiological aspects of Chapter 3, which significantly enhanced the quality of this study.

I would like to thank the Waterloo Algebraic Intelligence & Computation Lab members, Dr. Amena Assem, and Professor Attila Egri-Nagy, as well as the members of my thesis committee, Professor Jeffrey Shallit and Professor Sue Ann Campbell, for their valuable feedback, corrections, and suggestions, which greatly improved the thesis.

I gratefully acknowledge the financial support provided by a Mitacs Globalink Graduate Fellowship and the Natural Sciences and Engineering Research Council of Canada (NSERC), funding reference number RGPIN-2019-04669, and NSERC Special Fund for Trainees (Ukraine).

I am deeply thankful to my parents for their unwavering support and encouragement.

Dedication

To my parents, Yevheniia and Oleksandr.

Table of Contents

Author's Declaration	ii
Statement of Contributions	iii
Abstract	iv
Acknowledgements	v
Dedication	vi
List of Figures	x
List of Tables	xiii
1 Introduction	1
2 Deterministic-Probabilistic Finite Automaton: Algorithm and Examples	3
2.1 Introduction	3
2.2 Concept and context of terminology	6
2.2.1 CSSR algorithm	7
2.2.2 Proposed n -gram-merging algorithm	8
2.3 Constructing the n -gram machine	9
2.4 Minimization procedure	11

2.4.1	Equivalence of states	11
2.4.2	Determinism of transitions	13
2.4.3	Minimization procedure	15
2.5	Distance metrics	16
2.6	Existence of permutation subgroups	17
2.7	Examples of automaton reconstruction	20
2.7.1	Experiment setup	20
2.7.2	The even process	22
2.7.3	The length three cycle	24
2.7.4	The alternating group	25
2.8	Conclusion	26
3	Application of Finite Automaton to Distinguish Sequences of EEG Microstates	28
3.1	Introduction	28
3.2	Methods	31
3.2.1	Participants and experimental paradigm	32
3.2.2	EEG microstate computation	32
3.2.3	Distinguishing EEG microstates sequences with ϵ -machines	34
3.2.4	Linear separability of sets of epsilon machines after dimensional reduction	37
3.2.5	Strict separability of sets of epsilon machines without dimensional reduction	38
3.2.6	Methods validation using permuted data	40
3.2.7	Identifying n -grams with the biggest contribution to separation of sets of epsilon machines	40
3.3	Results	42
3.3.1	Separating participant groups: meditators and cotrols	42
3.3.2	Separating cognitive modes in meditators	50
3.3.3	Separating cognitive modes in non-meditators	54
3.4	Discussion	56

4	The Study of the Transformation Semigroup of the Abelian and Directed Non-Abelian Sandpiles	64
4.1	Introduction	64
4.2	Preliminaries	66
4.3	Abelian sandpiles	70
4.3.1	Model description	70
4.3.2	One-dimensional case: 2-regular non-directed graph	71
4.3.3	Complexity of Abelian semigroups	75
4.4	Non-Abelian sandpiles on the rooted trees	77
4.4.1	Model description	77
4.4.2	Embedding of the sandpile with equal to one capacity	78
4.4.3	Generalization for a rooted tree with arbitrary capacity	81
4.4.4	Aperiodic complexity of sandpile semigroups	82
4.5	Conclusion	83
5	The Summary of Contributions and Further Work	84
	References	86

List of Figures

2.1	The even process.	22
2.2	Reconstruction of the probability distribution over strings of length $k = 4$ measured by Hellinger distance. The quality is shown in percentages, where the highest value reflects zero distance between distributions. The left column corresponds to the CSSR algorithm and the right to the n -gram merging method. Every cell of the table corresponds to the algorithm's parameter configuration: the row denotes the history length (from 2 to 8) and the column denotes the significance level(for CSSR) multiplied by 10^2 or the minimization threshold Δ (n -gram merging).	23
2.3	The result of the group reconstruction by CSSR algorithm over the considered parameter range: row denotes the history length (from 2 to 8) and column denotes significance level multiplied by 10^2 . The color reflects the presence of subgroups in automaton's transformation semigroup: red - aperiodic (no non-trivial subgroups), light green - C_2 , yellow green - C_3 , dark green - C_4 , aqua green - C_6	24
2.4	The C_3 cycle.	25
2.5	The alternating group.	26
3.1	The schematic representation of computation of EEG microstate sequences [31].	33
3.2	The topographical maps for canonical EEG microstate classes A, B, C , and D [43] maps from the KeyPy template library (reduced from 64 to 30 channels and rendered with the EEGLAB topoplot function).	34

3.3	Work pipeline, explaining how EEG microstate sequences of two participant groups and three mind modes are analyzed and compared. A) - Group-level ϵ -machines are constructed from combined individual recordings for a range of parameters. B)- For a set of ϵ -machines the question of the group separability and mind-mode separability within the groups is addressed, using two types of criteria: B-I) Linear separability in 2D projection of the Euclidean embedding of the distance metrics; and B-II) strict separability criteria applied to original distance metrics. These criteria capture different types of group-level machine layouts relative to each other in metric space. C) - Successful separations of groups and mind modes for different parameter configurations are explained with the sets of n -grams having the biggest contribution to distance between sets.	36
3.4	The result of differentiating participant groups: meditators and non-meditators, using linear separation criteria using epsilon (left) and Jaccard (right) distance metrics for three temporal scales (clock, peak event). The green and red colors reflect the success and failure of separating group-level epsilon machines respectively.	43
3.5	2D projection of Euclidean embedding for n -gram machines (first line of figures) and minimized machines (second line) for selected cases of two-sided separation by strict criterion. The first and second columns represent the case of clock time mode, using epsilon distance type, where minimization is done with $\delta = 0.05$, history length $L = 4$ (first column), and history length $L = 5$ (second column). The third column represents the case of peak time mode, using jaccard distance variation, history length $L = 3$, where minimization is done with $\delta = 0.15$	44
3.6	The distribution of separation percentage between meditators and non-meditators comparing observed and permuted data.	46
3.7	The separation percentage between meditators and non-meditators over parameter range.	48
3.8	The success of linear separation of cognitive modes within meditators - panel A, and non-meditators - panel B, using 6 group-level machines over parameter range. Colors within each cell denote linear separability of mind mode: purple - mind-wandering, green-verbalization, yellow-visualization.	51
3.9	The distribution of separation percentage of cognitive modes within meditators - panel A, and within non-meditators - panel B, comparing observed and permuted data.	53

3.10	The success of strict separation of cognitive modes within non-meditators, using 6 group-level machines over parameter range. Colors within each cell denote linear separability of mind mode: purple - mind-wandering, green-verbalization, yellow-visualization.	57
4.1	The structure and multiplication table of flip-flop transformation semigroup.	67
4.2	The general structure of cascade transformations $w = (w_1, w_0) \in W$ of the wreath product $FF \wr FF = (Y, W)$ of two flip-flop semigroups $FF = (X, S)$, where $Y = X \times X$. The example of such transformation $t \in W$ is given in cascade representation and list transformation notation, where $t_i = t(y_i)$ for lexicographically ordered states $y_i \in Y$	68
4.3	The illustration of the structure of the identity elements and generators for a circle graph with $N \in \{2, 3, 4, 5\}$ non-sink vertices. The color indicates the number of sand grains in the corresponding vertex, and the number in the vertex denotes the ordering. The blue color denotes an empty vertex, the green denotes one containing a single sand grain, and the gray vertex is a sink.	75
4.4	The sandpile on the path graph $\mathcal{G} = (V, E)$ with $ V = N+1$ labeled vertices. The root vertex drawn in black is the sink, every vertex can contain up to 1 sand grain.	79
4.5	Example of the extended tree. Labels on the vertices refer to their capacities. Segments of the binary vector, corresponding to the extended vertices match the colors. The example shows the equivalence of the permutations within the blocks.	82

List of Tables

3.1	Successful cases of strict separation between meditators and non-meditators.	45
3.2	Statistical measures for two-sided separation percentage between groups. .	47
3.3	Selected cases of strict separation with highest separation ratio (i) between control (C) and meditator (M) groups, (ii) between a cognitive mode (mind-wandering (MW), verbalization (Ver), visualization (Vis)) and the other two modes within groups. Parameters are abbreviated for <i>time scale</i> : clock c, peak p and event e; values following ‘n’ and ‘d’ correspond to the <i>n-gram length</i> and <i>minimization threshold</i> Δ ; <i>distance metric</i> : last letter j stands for Jaccard and e for epsilon. I_1 denotes the block of <i>n</i> -grams (intersection of causal states) having largest contribution to distance between separated sets of ϵ -machines; its regular expression is marked by \star in the ‘Causal State’ column (in the selected cases, this intersection always coincides with one of the causal states). I^* denotes the union of the blocks with largest contributions to distance accounting for at least 10% of distance. A dot ‘.’ denotes any letter, successive dots can be different letters. \dagger can never have two of the same letters appear consecutively. A dash ‘-’ replaces regular expressions that are too long.	49
3.4	One-sided separation of cognitive conditions within the meditators group. .	52

Chapter 1

Introduction

The dynamical systems with discrete state space and event-driven state transition mechanisms often exhibit stochastic behavior. Modeling such systems effectively requires tools that can handle not only sequence generation and state transitions but also uncertainty and randomness. Among the variety of mathematical models, the probabilistic finite automaton provides a good framework for understanding the complex structure and behavior of such systems. The probabilistic nature of state-based structure allows us to estimate the likelihoods of systems trajectories, and the labeled representation of local state transitions allows us to capture the global emergent behavior through the analysis of the corresponding transformation semigroup. Although the probabilistic finite automata are associated with stochastic regular languages, within this work they are considered in the most general sense as a model of a stochastic event-driven system, without concentrating on their grammatical properties. In a practical setting, the automaton structure is not predefined but inferred from the available observations of the system processed by some algorithm. Then analysis of statistics and transformations of the machines would be performed in a computational manner by comparing probability distributions in a metric space and scanning for the presence of permutation structures. Additionally, the algorithm itself can be analyzed for efficiency, accuracy, or its power to recognize certain system's properties. However, for a more detailed theoretical analysis of the transformation semigroup of an automaton, one should operate with its well-defined algebraic representation. Therefore to explore finite automata from both theoretical and practical sides two different models are considered in this thesis, with an emphasis on the different aspects.

The first model is a deterministic probabilistic finite automaton (DPFA) considered through two chapters. Chapter 2 describes the algorithm for reconstructing the automaton from the sequences of discrete observations. The presented study does not concentrate on

the performance optimization or theoretical guarantees of the algorithm, but includes a practical implementation of the method and its experimental evaluation on three case studies, considering both the quality of reconstructing the probability distribution over strings and the ability to reconstruct the permutations of system's states from the data. Chapter 3 develops the framework for applying the described finite automata model to the analysis of EEG microstate sequences, offering insights into the potential of automata-based techniques in neuroscience. The studied question is distinguishing the participant groups and cognitive modes by comparing inferred from EEG data machines in a metric space.

The second model is an abstract sandpile model on finite directed graphs considered through Chapter 4 from a theoretical perspective. In its original formulation, the model was proposed by physicists as a dynamical system on a cellular automaton displaying a self-organized criticality phenomenon; later it was mainly studied in the field of algebraic graph theory. Although this model comes from different applications and has a deterministic nature, it shares structural parallels with automata through its transformation semigroup. The study of the usual Abelian version of the model is considered for a simple circle graph and includes the exploration of the identity and generator configurations and semigroup complexity. The study of the non-Abelian modification of the model for the rooted trees includes exploration of the semigroup by constructing its wreath product decomposition and understanding its aperiodic complexity.

Chapter 5 concludes the presented study of practical and theoretical aspects of finite automata models by summarizing the contributions and outlining the possible further direction of the study.

Chapter 2

Deterministic-Probabilistic Finite Automaton: Algorithm and Examples

A finite-state machine is the simplest model to describe the states and transitions of the system or the process, that can help to understand its structure and behavior. A deterministic-probabilistic automaton can be learned from the sequences of discrete observations, approximating the probability distribution over the observed/generated strings and giving the understanding of the system's structure, i.e., absorbing/terminal states, permutations or reversibility of transitions, etc. As the original system is unknown, both the structure and the underlying transition distribution have to be learned from the data. This chapter describes the details of automaton structure, a new reconstruction algorithm, and its implementation, as well as experimentally considers several examples.

2.1 Introduction

A finite state automaton is an abstract machine that consists of a finite set of states and labeled directed edges denoting transitions for every input symbol. Considering the output produced during the transition process there are two main types of machines: acceptors, returning a binary result, and transducers, returning the output sequence. In acceptors, every state has a binary type (accepting or not), and for every input string, a binary verdict on its acceptance is the type of the resulting state. A set of all accepted strings is called the regular language recognized by the automaton. The transducers at

every transition emit the symbol from the output alphabet, which is determined by the visited state (Moore machines) or by the pair of visited state and received input (Mealy machines). Another classification of machines is whether the successor state for every pair of current state and input symbol is uniquely defined (deterministic), or not (non-deterministic). A deterministic probabilistic (or stochastic) finite automaton (DPFA), is a generalization of finite state machine, where for every state, there is a probability distribution over transitions from that state. The probability distribution over the input alphabet describes the likelihood of every letter to be received at the current state, but once it is determined, the successor state is unique, that is, a machine is deterministic in the sense of transitions. The letters of the input alphabet can be understood as events happening in the system and driving the transitions, thus it is an event (input)-driven system. The DPFA can be seen as a generalization of a Markov chain, where transitions are also probabilistic but there is no input and the process is state-driven. While Markov chains are generally used to model stochastic processes, random walks, or time series, the main application of the DPFA can be seen in stochastic language recognition and sequence likelihoods. If an automaton has a designated start and a set of accepting states, then all words accepted by it describe a stochastic regular language, where every word is assigned a non-zero probability of being accepted. Another (equivalent) terminology for transition labels would be to, instead of input, perceive them as output that the system emits transitioning between the states, available as the observations of the system’s dynamics. Then such a DPFA can be interpreted as a generative model of a discrete event stochastic dynamical system.

The problem of reconstructing the DPFA from observation sequences can be formulated as learning the stochastic regular language or approximating the probability distribution over strings, and it was considered in the framework of probably approximately correct (PAC) learning. The goal is to construct an efficient algorithm (e.g., polynomial complexity) that with high probability returns a hypothesis with low error for any or a certain class of models. As the observed frequencies of strings would correspond to a perfect estimate, such conditions can be added as a limit on the number of states and the ability of generalization (being capable of generating strings that were not observed in data). Learning the automaton ‘in the limit’ assumes bounded growth of hypothesis size, i.e., adding more data can only result in a finite number of changes to the learned model. For example, [19] show that DPFA cannot be learned in the limit with probability one from only data of polynomial size, and they become PAC-learnable for L_∞ norm; [54] suggested the polynomial time algorithm for PAC-learning the distributions with acyclic DPFA for Kullback-Leibler divergence as the error measure, later [14] for general DPFA, and [49] for variation distance. The comparison between some methods for learning probabilistic automata was performed in a competitive manner during the PAUTOMAC competition [67]. For example, the team Shibata and Yoshinaka [59] compared three techniques: the

modification of ALERGIA [11] baseline method, a variable-length gram method, and a collapsed Gibbs sampling (CGS) method to estimate the DPFA, empirically showing that CGS performs the best, and is the overall winner of the competition.

There are also other models of stochastic systems that have a lot of similarities in structure and functionality with the described DPFA. Two examples are partially observable Markov decision processes (POMDP) [30] and Hidden Markov Models (HMM) [51]. The POMDP are associated with online learning problems in artificial intelligence, where the agent has no prior knowledge about the environment or its states but receives the observations and rewards after performing some actions and aims to learn the optimal response strategy. The POMDP are used to model responses of stochastic environment and similarly to DPFA, the agent’s actions are the input and both the probabilistic transitions and the output (consisting of reward and probabilistic observations) depend on current (unobserved) system’s state and received input. Then the agent, who only has access to observations and rewards, updates its belief about the environment and best response policy. To account for histories, alternatively, to such methods as recurrent neural networks, learning of finite automaton model that classifies histories into causal states can be utilized. Another commonly used structure to model observed stochastic processes is HMM, used in many applications, for example, signal processing [15] and pattern recognition [39]. In this model, every system’s state has two probability distributions: over states and over alphabet symbols. As in the Markov chain, the probabilistic transitions are not input-driven by depend only on states, however, the states are hidden and emit the observable symbols. The emission of output depends on the current state’s distribution over symbols but is independent of the triggered transitions. The algorithms to work with HMMs, for example, the Baum–Welch algorithm [6] to estimate probabilities of linear chain HMM or the Viterbi algorithm [68] to find the most likely path of hidden states, both consider the system’s structure predetermined but not inferred from data. While there are various methods available for modeling stochastic dynamical systems, the presented study only focuses on one specific DPFA reconstruction technique.

In this chapter, the state-merging algorithm for learning the DPFA model from the observation sequences based on n -grams is described and also implemented for further assessment and application.¹ However, there is another method, the causal state splitting reconstruction (CSSR) [57] technique, that is also based on n -grams and was presented as a powerful tool for inferring the causal structure of the observed system, the motivation for the method introduced and utilized here was to consider the simplest opposite (based on merging rather than splitting) technique. For experimental performance evaluation within

¹It is not suggested as a new design for performance improvement and such properties as efficiency or error bounds are not analyzed, also it is not compared to other models mentioned above.

the scope of this study three case studies are considered and compared to CSSR in such aspects as reconstructing the probability distribution over strings and permutation groups. In Chapter 3 it is evaluated for neuroscience applications.

2.2 Concept and context of terminology

Considering such an automaton as the computational model of a physical system [17], Crutchfield emphasized the connection of the inferred system to the measuring instruments used to obtain the sequences of discrete observations. The exact states of the original system are unknown, and the parameterized partition of the observed state-space visited over time is constructed to describe the observations. Every letter of the alphabet denotes a partition element, which is parameterized by the particle size ϵ and the time interval τ . The influence of the measuring instruments on the produced sequences further impacts the inferred deterministic-probabilistic finite automaton. Therefore, Crutchfield proposed to call the reconstructed approximate automaton ϵ -machine to highlight this influence. Although the stages of collecting the observations and constructing the automaton are usually independent, and the reconstruction algorithms generally do not directly take into account the measuring instrument and such parameters as ϵ , the above terminology was adopted in this work. Such expressions as ‘deterministic-probabilistic finite automaton’, ‘ ϵ -automaton’, and ‘ ϵ -machine’ further in the manuscript are equivalent and refer to the same notion throughout the text.

Let w denote the sequence of observations $a_i \in A$ of the dynamical system, and for any specific moment of time t , denote the history up to that moment t as \overleftarrow{w}_t , that is:

$$w = \dots, a_{t-2}, a_{t-1}, a_t, a_{t+1}, a_{t+2}, \dots, \quad \overleftarrow{w}_t = \dots a_{t-2}, a_{t-1}, a_t.$$

Denote by $\overleftarrow{\mathbf{W}}$ the set of all possible histories, and by $\overrightarrow{\mathbf{W}}$ the set of all possible futures $\overrightarrow{w}_t = a_{t+1}a_{t+2}\dots$ over the input alphabet A .

Definition 2.2.1. [58] *Consider the partition of histories $\overleftarrow{w} \in \overleftarrow{\mathbf{W}}$ into classes $\overleftarrow{\mathbf{W}}/\equiv$, such that two histories belong to the same class if and only if the corresponding conditional probability distributions over future trajectories of the system are the same:*

$$\overleftarrow{w}_1, \overleftarrow{w}_2 \in [\overleftarrow{w}] \in \overleftarrow{\mathbf{W}}/\equiv \Leftrightarrow \forall \overrightarrow{w}^* \in \overrightarrow{\mathbf{W}} P[\overrightarrow{w}^* | \overleftarrow{w}_1] = P[\overrightarrow{w}^* | \overleftarrow{w}_2].$$

This is the minimal homogeneous partition of histories, and its classes are called causal states of the process.

The above definition theoretically considers the infinite histories and futures. However, the idea of such formulation is that the corresponding partition function that sends the histories to equivalence classes $\overleftarrow{w} \mapsto [\overleftarrow{w}]$, ideally should not be limited to any finite memory, but capture the causal effects in the observational sequence.

So far it was mentioned that the procedure of the DPFA reconstruction concentrates on (i) inferring the structure (states and transitions) and statistics (transition probabilities) of the deterministic finite automaton, when also (ii) approximating the probability distribution over the generated strings. The introduced concept of causal states when utilized in the context of DPFA reconstruction also highlights the aspect of recognizing the temporal patterns in the data (what histories form the causal states) that capture non-limited memory causal effects in the system. Following the preferred terminology, the states of the reconstructed automaton are referred to as causal states in the manuscript.

2.2.1 CSSR algorithm

The causal state splitting reconstruction algorithm (CSSR) [57, 58] for inferring the structure of ϵ -machine from the observations of the system was introduced as a new algorithm for pattern discovery in time series that converges in a number of states and complexity ‘from below’. The state-merging algorithms, converging to the model ‘from above’, start from the maximal number of states that are the best fit to the data and proceed by combining them based on similarities. As opposed to it, CSSR starts by assuming the minimal complexity - only one causal state with the self-looped transitions, independent and identically distributed over the alphabet. After the system is initialized, the next stage of homogenization starts. Iteratively increasing the currently considered length of the history, it proceeds by splitting the histories into different causal states when the corresponding distribution over the alphabet has a significant statistical difference from the current estimate of transition probabilities from the causal state. The initialization corresponds to the zero history (empty suffix) and homogenization ends when the histories of maximum length n are considered, where n is a parameter. Thus in the resulting construction, the causal states are actually the classes of length- n substrings (contiguous blocks of n letters), later called n -grams. The next and last stage of the algorithm is determinization, which removes all transient states from the system and does further partition of states until transition determinism is achieved. The resulting causal states of the inferred system are claimed to be the minimal states that have a homogeneous distribution for the next symbol and are deterministic.

The resulting automaton is Markovian with respect to causal states, but not with respect to the observation sequences. The algorithm was analyzed in more detail in [58],

considering its time complexity, the convergence of the empirical conditional probabilities, and the kinds of statistical errors it can produce. The original implementation of the CSSR by K. Klinkner, C. Shalizi, and J. Crutchfield is available open-source on GitHub [56].

2.2.2 Proposed n -gram-merging algorithm

When the CSSR algorithm was introduced it was claimed that it should do better than other approaches in terms of reconstruction accuracy and rate of convergence, also capturing the unbounded temporal dependencies in data. However, it was not explicitly placed within the framework of all other existing DPFA reconstruction techniques by comparing their properties, but in some sense stayed aside because of being proposed from the physics rather than computer science perspective. Although the CSSR method and its application to the EEG data were the main starting point and reference prior to this study, the scope of this thesis does not include the improvement of its performance or explicit comparison of it to other approaches.

The approach considered in this thesis falls into a different category of state-merging algorithms. Similarly to CSSR, it is based on classifying the n -grams to the equivalence classes based on the empirical transition probabilities, but differently (i) it starts with every n -gram in a separate state and merges them, instead of starting with all n -grams in one state and splitting them, and (ii) to decide on a similarity of two probability distributions it requires the distance between them to be bounded by threshold Δ instead of using a statistical test, and (iii) the determinization stage of CSSR does the further splitting of the states, while considered method proceeds by the further merging of states.

Among the other DPFA reconstruction methods, for example, the well-known baseline state-merging algorithm ALERGIA [11] starts by building the prefix-tree acceptor of the sample data and uses a statistical test (as CSSR does) to decide on similarity of nodes. In the n -gram language modeling the smoothing techniques that adjust empirical probability estimates are often used to improve the performance [13]. Another n -gram-based method that classifies words into classes based on conditional transition probabilities of future trajectories, is called topological merging [50]. Although mentioning the notion of Δ -close transition probabilities they considered a simplified approach $\Delta = 1$ requiring equality of the distribution support regardless of the probabilities. As it is missing the determination stage (by either merging or splitting), it was suggested to recover causal states, but not the complete DPFA.

The algorithm considered in the thesis does not explicitly follow the exact description of any encountered methods; however, all of the algorithm's decisions occur in some form

in the works referred to above. Thus, it is not positioned as a novel, previously unutilized technique, but just one of the possible DPFA reconstruction methods. The rest of this chapter describes the algorithm and considers some of its properties, and the next chapter conducts its application to distinguishing EEG microstate sequences.

2.3 Constructing the n -gram machine

The deterministic-probabilistic automaton (ϵ -machine) to be constructed is described by a 4-tuple $\mathcal{E} = (S, A, \delta, p)$ where the components are:

- a finite set of states S ,
- a finite alphabet A denoting events occurring in the system,
- a transition function $\delta : S \times A \rightarrow S$ mapping pairs of the current state and occurred event to the successor state,
- a probability function $p : S \times A \rightarrow [0, 1]$, $\forall s \in S$, $\sum_{a \in A} p(s, a) = 1$, describing for every state the probability distribution over the events' alphabet, which maps state-event pairs to the conditional probability $P(a|s)$.

Additionally, a probability distribution $S_0 : S \rightarrow [0, 1]$ for the start state can be defined. The input (event) alphabet is defined by all letters occurring in the sequences of observations. The set S of states is the set of causal states of the system (equivalence classes of histories having the same probability distribution over all possible futures) extracted from observations. In practical settings, instead of dealing with infinite histories and futures, the n -grams are used.

The first stage of the construction process is to build a n -gram machine \mathcal{E}^* , where n is a parameter. Assume that every history leads the system to a different state, that is, the state space $S = A^n$ consists of all possible n -grams over the alphabet A .

The transition function from the state $w = w_1 \dots w_n \in S$ with the input letter $a \in A$ is defined as

$$\forall w = w_1 \dots w_n \in S, w_i \in A, \forall a \in A, \delta(w, a) = w_2 \dots w_n a \in S,$$

that is, the successor n -gram after observing new event a is defined by appending the new observation to the current state w and forgetting the oldest letter w_1 that was remembered in w .

The probabilities of corresponding transitions are defined as their empirical frequencies in the observation sequence:

$$\forall w \in S, a \in A, \quad p(w, a) = \frac{\# \text{ of occurrences of } wa}{\sum_{x \in A} (\# \text{ of occurrences of } wx)},$$

taking into account only occurrences of the n -gram w followed by a letter.

Note that there are n -grams that never occur in the data or that only occur at the end of the observation sequence. In the latter case, the information about the future after such states are absent for unknown reasons, as it may indicate that a special terminal state of the system was visited, the presence of noise, or just the termination of the recording. Therefore, when constructing the n -gram machine, consider the following types of states (n -grams):

- *Impossible*, I —states that do not occur in the data,
- *Real*, R —states that occur in the data at least once,
- *Dead*, D —states that only occur at the end of lines,
- *Unknown*, U —the state into which all transitions from every dead state lead—an additional state to describe the uncertainty that follows after reaching dead states.

For the states from $R \setminus D$, the transition and probability functions were already defined above. For the states of types I, D, U , the following modifications are introduced to complete the automaton's description:

1. All non-occurring n -grams are combined into one state I since they are all equivalent and we do not need to store any transition information about them separately.
2. For all events, the transitions from the state I lead to I (are self-loops) with equal nonzero probabilities. Their values do not matter since all edges entering I have zero probability.
3. All transitions from state D lead to state U with equal probability.
4. All transitions from state U are self-loops with probabilities equal to the frequencies of corresponding letter in the input data.

Additionally, for all $w = w_1 \dots w_n$ and for all $a \in A$ if $w_2 \dots w_n a$ occurs in the data, but $w_1 w_2 \dots w_n a$ does not, then the corresponding transition $\delta(w, a)$ connects two states of type R , but will have zero probability $p(w, a) = 0$. When considering the transformations that events A describe on the states S , such impossible transitions will influence the corresponding transformation semigroup distorting the actually observed information about the system's transformations. Therefore all zero probability edges from states of type R are reset to state I . This completes the construction of the n -gram automaton, also mentioned later as a pre-epsilon machine.

2.4 Minimization procedure

The constructed n -gram machine reflects all transitions observed in data and approximates the probability distribution over the strings. Generally, the size of this model grows exponentially in n , the structure only depends on the occurrence of words in data, which in the worst case is the 'full-automaton', and there is a high possibility that two distinct n -grams representing different states are actually almost indistinguishable in terms of predicted future trajectories of the system.

The minimization procedure aims to reduce the size of the automaton, reducing redundancy in the number of states, but preserving the distribution over generated sequences; to detect what n -grams are similar in statistical/causal sense and potentially reveal the aspect of the underlying system's structure.

2.4.1 Equivalence of states

The description given in Section 2.2 of the causal states theoretically requires the equality of probability distributions for all possible trajectories of the system. Relaxing this requirement to finite futures the following definition implies

Definition 2.4.1. *Two states $X, Y \in S$ of epsilon automaton $\mathcal{E} = (S, A, \delta, p)$ are considered equivalent if any future trajectory of up to N steps is equally probable for them. That is,*

$$X \equiv Y \iff \forall f \in A^+ \text{ with } 1 \leq |f| \leq N, P(f|X) = P(f|Y).$$

The following Lemma shows that for such an equivalence condition, it is enough to consider only trajectories of the maximal length N .

Lemma 2.4.2. For two states $X, Y \in S$ of the epsilon machine $\mathcal{E} = (S, A, \delta, p)$, let P_X^L and P_Y^L denote the probability distributions over futures $f \in A^L$ of length L over the alphabet A , respectively. Then the following holds

$$P_X^{L+1} = P_Y^{L+1} \implies P_X^l = P_Y^l, \text{ for all } 1 \leq l \leq L.$$

Proof. For any future $f \in A^L$ and $a \in A$ the recursive probability definition gives

$$P_X^{L+1} = P_Y^{L+1} \implies P(f|X) \cdot P(a|X \cdot f) = P(f|Y) \cdot P(a|Y \cdot f),$$

where $X \cdot f$ and $Y \cdot f$ denote the state reached from X and Y after applying transitions of f respectively. Considering the sum over all events in the alphabet, we obtain

$$\sum_{a \in A} P(f|X) \cdot P(a|X \cdot f) = \sum_{a \in A} P(f|Y) \cdot P(a|Y \cdot f) \implies P(f|X) = P(f|Y),$$

due to the following property of epsilon-machines

$$\sum_{a \in A} P(a|s) = 1, \forall s \in S.$$

Recursively repeating the above reasoning for arbitrary $f \in A^L$, we obtain the equality of probability distributions $P_X^l = P_Y^l$, for all $1 \leq l \leq L$.

□

In practical settings, the requirement on the exact equality of probabilities has to be relaxed, especially given that transition probabilities are estimated from the data and such a definition is sensitive to slight changes in the data stream. The two commonly used approaches are to use the statistical test for equality of two distributions, or to consider equality within the allowed deviation threshold. Both choices introduce an additional minimization parameter. In this study the second approach is considered, which requires the following definitions.

Definition 2.4.3. Two states $X, Y \in S$ of the ϵ -machine $\mathcal{E} = (S, A, \delta, p)$ are Δ -close if the distance between them does not exceed the allowed threshold: $X \sim_{\Delta} Y \Leftrightarrow d(X, Y) \leq \Delta$.

Definition 2.4.4. The **distance** $d(X, Y)$ between the states $X, Y \in S$ of the ϵ -machine $\mathcal{E} = (S, A, \delta, p)$, constructed for the n -gram length N is defined based on the probabilities of future trajectories of length N from states X and Y in several ways:

1. as the maximum absolute difference between the distributions over futures $f \in A^N$:

$$d(X, Y) = \max_{f \in A^N} |P(f|X) - P(f|Y)|,$$

2. as the cumulative absolute difference between the distributions over futures $f \in A^N$:

$$d(X, Y) = \sum_{f \in A^N} |P(f|X) - P(f|Y)|,$$

3. as the maximum absolute difference between one-step transition probabilities over the trajectories $f = f_1, \dots, f_N \in A^N$ of transitions:

$$d(X, Y) = \max_{f \in A^N, 1 \leq i \leq N} |P(f_i|X_i) - P(f_i|Y_i)|, \quad Z_1 = Z, Z_i = \delta(Z_{i-1}, f_{i-1}), \quad Z \in \{X, Y\}.$$

The above \sim_Δ relation is not transitive, so it does not define a partition of states into equivalence classes. Therefore we define the Δ -equivalent relation \equiv_Δ as the transitive closure of the \sim_Δ relation:

Definition 2.4.5. *Two states $X, Y \in S$ of the ϵ -machine $\mathcal{E} = (S, A, \delta, p)$ are Δ -equivalent $X \equiv_\Delta Y$ if and only if there exists a chain of Δ -close (\sim_Δ) states between X and Y :*

$$\forall X, Y \in S, \quad X \equiv_\Delta Y \Leftrightarrow \exists \{X = Z_1, Z_2, \dots, Z_m = Y\} \subseteq S, \text{ s.t. } Z_i \sim_\Delta Z_{i+1}, \quad 1 \leq i < m.$$

The above relation is reflexive, symmetric, and transitive and is the simplest algorithmic decision that defines a partition of a machine's states into disjoint classes. However, it deviates from the original definition significantly and can result in very different states being associated with the same class along the chain, such that the maximum distance between the members of the same class cannot be estimated and is not bounded in the general case.

2.4.2 Determinism of transitions

The equivalence relation above does not guarantee the uniqueness of the successor state of the transition function:

$$\forall X, Y \in S \quad X \equiv_\Delta Y \not\Rightarrow \delta(X, a) \equiv_\Delta \delta(Y, a), \quad \forall a \in A.$$

The further recursive merging of the descendants of the Δ -equivalent states can be applied to achieve the transitions' determinism. However, while all of the *Real* descendants have to be merged into one class, the *Impossible* and *Unknown* states have to remain separated from all the *Real* states. Thus, it is necessary to abort the corresponding merge branches in the implementation, while also ensuring that the merge remains an order-independent function. Then, the transition from the classes is set to the child state of the highest priority in the order: (1) *Real*, (2) *Unknown*, (3) *Impossible*. As not every member of the class can give information about the successor state, the merge function has to guarantee the assignment of the proper representatives of the classes. Also, as the transition information is used for recursive calls, the transition table has to be updated simultaneously with the class labels. The pseudocode of the corresponding function is given as Algorithm 1.

Algorithm 1 Function MERGE(X, Y)

Input: States $X, Y \in S$, labeling function $[] : S \rightarrow \text{representatives } S' \subseteq S, X \mapsto [X]$, transition function $\delta : S \times A \rightarrow S$

Updates: labeling function, transition function

Ensures: determinism of highest priority transitions from X and Y for all real futures

```

1: if  $X, Y$  have same label  $[X] = [Y]$  then return
2: end if
3: Remember representatives  $rX = [X], rY = [Y]$ 
4: Assign same label  $[X] = [Y]$ 
5: for all events  $a \in A$  do
6:   if both transitions are real  $\delta(rX, a), \delta(rY, a) \in R$  then
7:     Merge( $\delta(rX, a), \delta(rY, a)$ )
8:   else
9:     Set transition  $\delta([X], a)$  to higher priority among  $\delta(rX, a)$  and  $\delta(rY, a)$ 
10:  end if
11: end for

```

The recursion ensures that the merge function will not exit until all finite future trajectories are considered and the function will be called non-trivially at most $|S|$ times. After two classes have received one label, they cease to be distinguishable for this function, since it always looks only at the representative of the class to which the element belongs. The proper information about the transition from classes will be stored for the representatives of the class, and the other entries of the transition table will be cleaned afterward, such

that the resulting transition function is

$$\delta([X], a) = \begin{cases} [\delta(X, a)] & \delta(X, a) \in R, \\ U & \delta(X, a) = U \text{ and } \nexists Y \in [X] : \delta(Y, a) \in R, \\ I & \forall Y \in [X] \delta(Y, a) = I, \end{cases}$$

for all $X \in R \subset S$ and for all $a \in A$.

Additionally, the consecutive calls of $Merge(X, Y)$ and $Merge(Y, Z)$ guarantees that all trajectories for X and Z are also merged, thus giving the order independence property.

2.4.3 Minimization procedure

From the above description, two states X and Y of an ϵ -machine will be assigned the same equivalence class if they are Δ -equivalent, i.e., there is a chain of Δ -close states between them, or they are descendants of two Δ -equivalent states along a *real* future trajectory.

Algorithm 2 Minimization procedure

Input: the n -gram machine $\mathcal{E}^* = (S, A, \delta, p)$, $n \geq 1$, threshold Δ

Output: minimized ϵ -machine $\mathcal{E} = (S, A, \delta, p)$

```

1: repeat
2:   for all pairs of states  $X, Y \in S$  do
3:     if states  $X$  and  $Y$  are  $\Delta$ -close then
4:       Merge( $X, Y$ )            $\triangleright$  recursively assigns  $X \& Y$  and  $\delta(X, a)$  &  $\delta(Y, a)$  to
5:                                $\triangleright$  the same class and defines deterministic transitions
6:     end if
7:   end for
8:   Identify states  $S$  with classes  $[X]$  of partition  $S / \equiv$ 
9:   for all states  $X \in S$  do
10:    Update probabilities of states and transitions
11:   end for
12: until there is no pair of  $\Delta$ -close states in  $S$ 

```

The Algorithm 2 gives a pseudocode of the minimization procedure and shows the main stages of partitioning the states into classes based on their similarities, recursively merging classes to guarantee the determinism of transition function for a given input letter, and updating transition probabilities between the resulting states. The resulting ϵ -machine is properly defined and follows the definition:

Definition. The ϵ -machine $\mathcal{E} = (S, A, \delta, p)$, is considered minimized for the minimization threshold Δ if no pair of its states are Δ -close.

The transition probabilities for equivalence classes are defined as previously, but accounting for the occurrence of all n -gram members of the class. Given the allowed distance Δ between the states, which also possibly accumulates along the chain of the merged states, the distribution over transitions from the class can differ a lot from the initial transition distribution of its n -gram member. While this averaging gives a more generalized model than over-fitted to data n -gram machine, it can also lead to undesired distortions of the reconstruction. Following the earlier defined terminology, the states of the minimized ϵ -machine are referred to as causal states later in the text.

2.5 Distance metrics

For the pairwise comparison of the epsilon machines $\mathcal{E}_1, \mathcal{E}_2$ with the same construction parameters two types of distance metrics are used: epsilon distance and Jaccard distance variation. Note that even if two epsilon machines with history length n have the same vocabulary, the partitions of n -grams into causal states are generally different. Therefore we extend the notation for the causal state of the history $w^* \in A^n$ above to $[w^*]_i$ to indicate that it refers to the epsilon machine \mathcal{E}_i .

Definition 2.5.1. The epsilon distance $d_E(\mathcal{E}_1, \mathcal{E}_2) \in \mathbb{R}$ is defined as follows:

$$d_E(\mathcal{E}_1, \mathcal{E}_2) = \frac{1}{|A|^n} \sum_{w \in A^n} \sum_{a_i \in A} |p_1([w]_1, a_i) - p_2([w]_2, a_i)|,$$

where n is history length, $[w]_i$ is the causal state to which the n -gram w is classified in \mathcal{E}_i , and $p_i([w]_i, a_j)$ denotes the empirical probability of observing event a_j after the causal state $[w]_i$ of the epsilon machine \mathcal{E}_i .

Definition 2.5.2. The Jaccard distance variation $d_J(\mathcal{E}_1, \mathcal{E}_2) \in \mathbb{R}$ is defined as follows:

$$d_J(\mathcal{E}_1, \mathcal{E}_2) = \frac{1}{|A|^n} \sum_{w \in A^n} \left(1 - \frac{\sum_{a_i \in A} \min(n_1([w]_1, a_i), n_2([w]_2, a_i))}{\sum_{a_i \in A} \max(n_1([w]_1, a_i), n_2([w]_2, a_i))} \right),$$

where $n_i([w]_i, a_j)$ denotes the number of occurrences of the event a_i after all of the n -grams w in the state $[w]_i$ in the epsilon machine \mathcal{E}_i .

Although all n -grams non-occurring in the data used for ϵ -machine construction are assigned to the ‘impossible’ state in machine \mathcal{E}_j , in the above expression for distances the corresponding counts and probabilities $n_j([w^*]_j, a_i) = 0$, $p_j([w^*]_j, a_i) = 0$ are set to zero, to not reduce the noticeable differences.

By definition of epsilon and Jaccard distances, they consist of differences in the probabilities of the corresponding transitions (epsilon) and differences in the number of occurrences of these transitions (Jaccard) accumulated over all states and events. The Jaccard distance is introduced to add sensitivity to how this transition is inherent in each machine. For example, suppose that for some n -gram $w \in A^n$ its equivalent histories $[w]_1$ in \mathcal{E}_1 occur 100 times, and the transition from this state occurs in 50 cases with event $a \in A$, that is $n_1([w]_1, a) = 50$, which means that the transition probability is $p_1([w]_1, a) = 1/2$. While in the second epsilon machine \mathcal{E}_2 , histories $[w]_2$ equivalent to w occur 1000 times where 500 times they are followed by even a , that is $n_2([w]_2, a) = 500$ which implies that transition probability is also $p_2([w]_2, a) = 1/2$. However, this transition in the first machine occurs much less frequently, which will be taken into account in the Jaccard distance through the difference in the number of occurrences of these transitions: $(500 - 50 = 450)$ instead of $(1/2 - 1/2 = 0)$ for the epsilon distance type.

2.6 Existence of permutation subgroups

The transitions between the states of ϵ -machine $\mathcal{E} = (S, A, \delta, p)$ are defined for all input events $a \in A$, each representing a transformation of the state space S . As every ϵ -machine is a deterministic finite automaton, without taking into account locally defined transition probabilities, one can consider a corresponding transformation semigroup, the algebraic structure that describes how the states of the automaton transition under the action of the input alphabet. The algebraic approach to analyzing an event-driven dynamical system concentrates on studying its transformation semigroup and involves understanding its structure by decomposing it into building blocks such as permutation subgroups. To do that we need a semigroup description of an unknown system as the only thing we can do is to observe it and collect measurements. Therefore, such an approach requires a stage of deriving the semigroup description from observations. This is what the automaton model does, but it will be especially interesting and useful if we can detect subgroups and this is why we raise this question in this chapter.

Definition 2.6.1. *The **transformation semigroup** T of a deterministic finite automaton $\mathcal{E} = (S, A, \delta)$ is a set of functions $f : S \rightarrow S$ generated by the $m = |A|$ transformations $a \in A$ denoted by the input letters and defined in the transition table. The transformations*

are closed under multiplication and associative. Every transformation $t \in T$ acts on the states $X \in S$ on the right, denoted as $X \cdot t \in S$ and can be defined as a finite non-empty word $w \in A^+$ over the alphabet A .

Definition 2.6.2. The transformation semigroup is called **aperiodic** if it does not have any non-trivial subgroups.

Lemma 2.6.3. The transformation semigroup $T(\mathcal{E})$ of the n -gram machine $\mathcal{E} = (S, A, \delta, p)$ does not have any nontrivial permutation group.

Proof. Suppose that there is a subset of states $X \subseteq S$ which is permuted by nontrivial group $G \subseteq T(\mathcal{E})$. Each permutation $g \in G$ is represented by the non-empty word $g = w \in A^+, w \neq \lambda$. The subset of states $X \not\subseteq \{I, U\}$ cannot consist only of *impossible* and *unknown* states because $I \cdot g = I$ and $U \cdot g = U, \forall g \in G$, but G is the nontrivial group. Hence X contains at least one *real* state.

Each *real* state $x \in X$ has *real* image $x \cdot g$. Because if $\exists g \in G : x \cdot g \in \{I, U\}$, then $\exists g^{-1} \in G : x = x \cdot (g \cdot g^{-1}) = (x \cdot g) \cdot g^{-1} \in \{I \cdot g^{-1}, U \cdot g^{-1}\} = \{I, U\}$ gives a contradiction. Consider the n -th power $g^n \in G$ of the permutation $g = w \in G, |w| \geq 1$ applied to the *real* state $x \in X$. As the successor state is *real*, by the transition function in the n -gram machine it will be defined as the suffix of the word $g^n = w^n, |w^n| = m \geq n$ of the last n letters of the word $g^n = w^n = w_1 \cdots w_m$:

$$\forall x = x_1 \cdots x_n \in S, \forall a \in A, \delta(x, a) = x_2 \cdots x_n a \in S \implies x \cdot g^n = w_{m-n+1} \cdots w_m \in S.$$

That is, $g^n \in G$ is a permutation of X that acts as a constant mapping on all *real* states of X . Hence X contains at most one *real* state, say x_0 . That is, $X \subseteq \{x, I, U\}$, and $\forall g \in G, \forall x \in X : x \cdot g = x$, therefore G is a trivial group. \square

Lemma 2.6.4. In the transformation semigroup of the minimized ϵ -machine $\mathcal{E} = (S, A, \delta, p)$, constructed from the n -grams of length n , the transformation $t \in T(\mathcal{E})$ represented by the word $t = w = w_1 \cdots w_m, m \geq n$ maps two *real* states $X, Y \in R \subset S, X \neq Y$ to two different *real* successor states $X \mapsto X \cdot t$ and $Y \mapsto Y \cdot t$, if the future trajectory w is *impossible* for all n -gram members of at least one of the states X or Y , but is *possible* for both causal states X and Y .

Proof. For the n -gram member x of the state X acted upon by the transformation $t = w = w_1 \cdots w_m$, consider the transition chain in the n -gram machine prior to minimization:

$$x_0 = x, x_m = x \cdot w, x_i = \delta(x_{i-1}, w_i),$$

if all transition probabilities along this chain are non-zero, every visited state is *real* and for $m \geq n$ the resulting state $x_m = x \cdot w$ is the n -gram $w_{m-n+1} \dots w_m$. As minimization guarantees that all descendants along the real trajectories are merged, the successor state for transformation t in the minimized machine is $X \cdot t = [w_{m-n+1} \dots w_m]$ the class of the corresponding n -gram. Similarly, $Y \cdot t = [w_{m-n+1} \dots w_m]$, contradicting $X \cdot t \neq Y \cdot t$. Thus, for $X \cdot t$ and $Y \cdot t$ to be distinct it is required that at least for one of the states X, Y the trajectory $t = w$ is impossible for all its n -gram members.

However, in the minimized machine the transition probability is zero if and only if the transition leads to the *impossible* state I , all transitions from which are the self-loops. This means that for the successor state $X \cdot t$ to be *real*, all states in the transition chain from X have to be real as well, giving the second requirement.

$$X_0 = X, X_m = X \cdot w, X_i = \delta(X_{i-1}, w_i), X, X \cdot w \in R \implies X_i \in R, 1 \leq i \leq m.$$

For the two above requirements to hold simultaneously, the following situation must occur. Without loss of generality, suppose that for the word $w_1 w_2$ of length two, there is an n -gram $x = v_1 \dots v_n \in X_0$ that gives transition to X_1 , that is $X_1 = [v_2 \dots v_n w_1]$. But for all n -gram members of state X_0 the transition with word $w_1 w_2$ is not possible, that is $\forall x = u_1 \dots u_n \in X_0$ there is no occurrence of $u_1 \dots u_n w_1 w_2$ in the data. Then for the transition to $X_2 = \delta(X_1, w_2)$ to have non-zero probability, there must be an n -gram in X_1 that is not a successor of X_0 's members, for which w_2 occurs afterward, i.e., $\exists v_1 \dots v_n \in X_1 \setminus \{u_2 \dots u_n w_1 \mid u_1 \dots u_n \in X_0\}$, such that the word $v_1 \dots v_n w_2$ occurs in data and $X_2 = [v_2 \dots v_n w_2]$. \square

The conditions of Lemma 2.6.4 are necessary for the transformation t to not act as a constant map on the subset of states. This implies that, for the existence of the non-trivial permutation group in the reconstructed (and minimized) ϵ -machine, many substrings must not occur in the observation sequences. From the definition of causal states, it follows that for two states to be recovered in reconstruction, their influence on the distribution of the generated strings has to be significantly different. The above observation suggests that for such structural properties of the system, as permutations of the states to be recovered from the generated sequences, the vocabulary generated from those states must also be different. In situations where the size of the input alphabet is very small and almost all words occur with non-zero probability, the above procedure for ϵ -machine reconstruction would only allow for recovery of the distribution over strings, but not the complex structural properties of the system.

2.7 Examples of automaton reconstruction

In the general case, the reconstruction of the automaton from the data is performed where no information on the original system is known. However, when the automaton that generated the data is available, it is possible to evaluate if the specific system's properties are preserved in the reconstruction. In the following example, we will consider how well the probability distribution over the generated strings is approximated and whether the automaton's transformation semigroup contains the permutation of the system's states.

2.7.1 Experiment setup

For every example $|S|$ observation sequences are generated from the reference automaton $\mathcal{A} = (S, A, \delta, p)$, giving a total length of $N = 30000$ symbols. Every state $s_i \in S$ of the automaton serves as a start state for the i^{th} sequence, each containing $N/|S|$ symbols. Once the start state is selected, the generating process keeps sampling the alphabet letters according to the distribution of the current state, and performing transitions according to the sampled symbols. The number that every state is visited during the simulation implies the probability distribution over states $Q^{\mathcal{A}}$ for reference automaton \mathcal{A} .

The hypothesis machine $\mathcal{B} = (S, A, \delta, p)$ is reconstructed from such data using two techniques: the n -gram-merging method described above and the CSSR algorithm [58], available open-source at GitHub [56]. Both algorithms also return the distribution over states $Q^{\mathcal{B}}$, which is calculated based on the occurrence of the n -gram members of the causal states in the observations.

Let $P_k^{\mathcal{A}}$ and $P_k^{\mathcal{B}}$ be the two probability distributions that automata \mathcal{A} and \mathcal{B} generate over all possible words of length k over the alphabet A , denoted as A^k , where k is the independent parameter. The corresponding distributions are calculated using the expectation of each word to be generated by the machine:

$$\forall w = w_1 \cdots w_k \in A^k, \quad \mathcal{E} = (S, A, \delta, p) \quad P_k^{\mathcal{E}}[w] = \sum_{s \in S} Q^{\mathcal{E}}[s] \cdot P[w|s],$$

where $P[w|s]$ is the probability of observing the trajectory w from the state s :

$$P[w|s] = \prod_{i=1}^k P[w_i|s_{i-1}], \quad \text{where } s_0 = s, s_i = \delta(s_{i-1}, w_i),$$

Then the goodness of the hypothesis \mathcal{B} with respect to \mathcal{A} is measured by the distance between the distributions $D(P_k^{\mathcal{A}}, P_k^{\mathcal{B}})$, where D can denote any distance metric or divergence. In later examples, the Hellinger distance $D_H(P, Q)$ will be used.

Definition 2.7.1. *The Hellinger distance between two discrete probability distributions $P = (p_1, \dots, p_n)$ and $Q = (q_1, \dots, q_n)$ is defined as*

$$D_H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^n (\sqrt{p_i} - \sqrt{q_i})^2}.$$

It is a bounded metric, the minimum distance zero is achieved if two distributions agree at every point of the domain, and the maximum distance 1 is achieved when P assigns probability zero to every set to which Q assigns a positive probability, and vice versa.

The above metric is also related to the total variation distance

$$D_{TV}(P, Q) = \frac{1}{2} \sum_{i=1}^n |p_i - q_i| \leq \sqrt{2} D_H(P, Q).$$

To illustrate the quality of the reconstruction for the considered examples, the distance $D_H(P_k^A, P_k^B)$ for the fixed value of parameter $k = 4$ is demonstrated in the percentage value, where the higher percentage denotes the smaller distance.

The transformation semigroup of every considered example contains a subgroup that permutes some states of the automaton. A second considered measure of the reconstruction quality is whether the corresponding permutation group is preserved in the reconstructed machine. To verify the above the Semigroup Decomposition (SgpDec) package [24, 23] of Groups, Algorithms, and Programming (GAP) [27] software was used. The semigroup of the automaton is described by the action of all generators (alphabet letters) on the states, and the verification returns one of the following verdicts for the transformation semigroup: (i) it is trivial, or (ii) it is non-trivial, but aperiodic (does not contain non-trivial subgroups), or (iii) it contains at least one non-trivial subgroup, specifying which exactly.

To assess the algorithms' performance the above measures were evaluated for the set of reconstructed machines for the fixed parameter range, same for all examples. The history length was varied from $n = 2$ to $n = 8$ with step one. The second parameter, significance level α for CSSR algorithm was considered from $\alpha = 0.003$ to $\alpha = 0.102$ with step 0.003; and minimization threshold Δ for n -gram merging method was considered from $\Delta = 0.005$ to $\Delta = 0.31$ with step 0.005. Two different parameter ranges were considered for the second parameter, as its nature and the role it plays are very different for considered algorithms. While it makes the comparison more reasonable, there is still some ambiguity in comparing their overall performance.

2.7.2 The even process

The dynamical system with two states and two transitions labeled with A and B , shown in Figure 2.1, describes the so-called even-process.

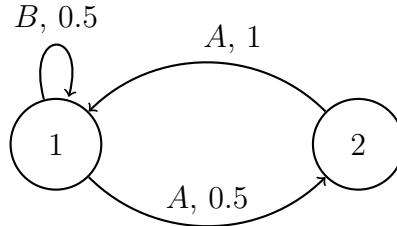
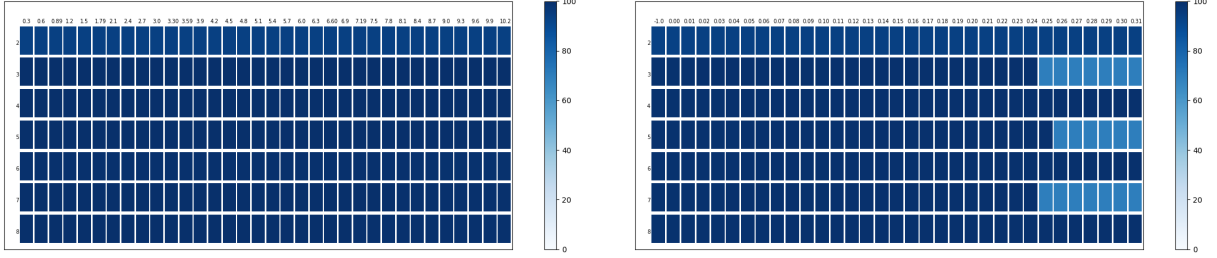


Figure 2.1: The even process.

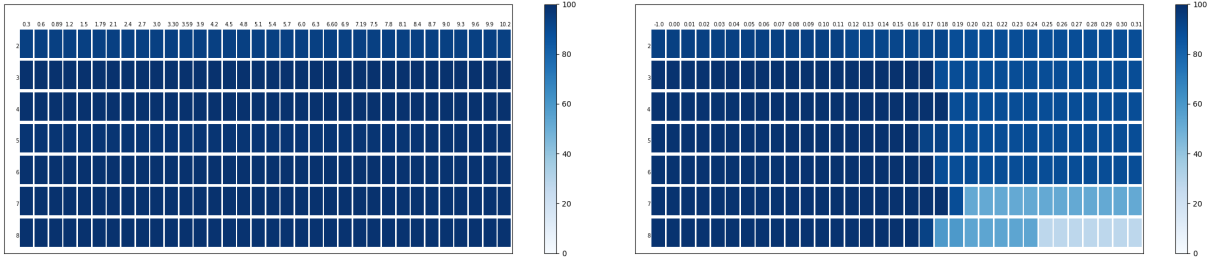
Both events are equally likely to be observed from the state 1, but only event A is possible from state 2, thus after A occurs once it is guaranteed it will be observed even number of times in a row.

Although the system is a first-order Markov process with respect to states, it is not Markovian with respect to generated sequences for any finite memory size. Despite being so simple to describe, it is sometimes called the infinite memory process, which becomes harder to reconstruct using finite memory when longer substrings are likely to occur (increasing the probability of A from state 1). Additionally, the transformation A in this process permutes the states of the system, generating the cyclic C_2 (counter) group.

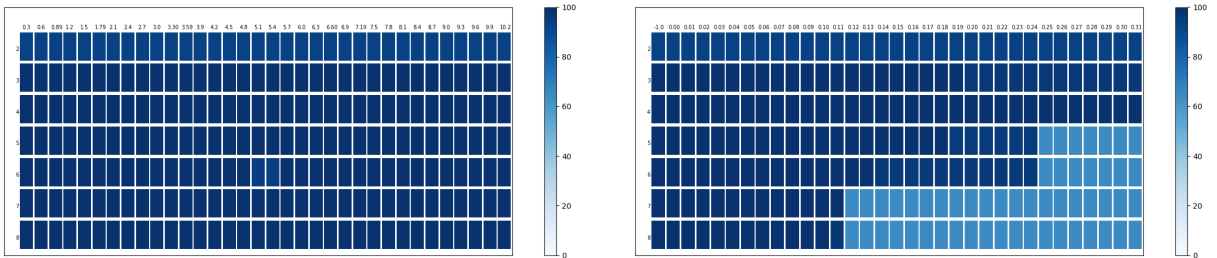
The quality of reconstructing the probability distribution is shown in Figure 2.2a for the CSSR on the left and for the n -gram merging method on the right. For the CSSR algorithm, the lowest reconstruction quality of 93.7% was achieved for history length $n = 2$, with only a slight difference in decimals when varying the significance level α . For the rest of the parameter range, it exceeds 99%, also without varying a lot with parameter change. For the n -gram merging method, the reconstruction accuracy shows the same result when the minimization threshold $\Delta \leq 0.24$. However, for $\Delta \geq 0.24$ there is a sensitivity to the parity of history length, showing a drop in accuracy to 70% for the odd values of n . Considering the transformation semigroup of the corresponding automaton, obtain that the drop in accuracy corresponds to the trivial semigroup. For the rest of the parameters of n -gram merging method, the semigroup corresponding to machines is non-trivial but aperiodic. However, the C_2 counter group was preserved in the reconstruction by the CSSR method for $n \geq 3$, as shown in Figure 2.3a. In some of the cases, a bigger cyclic group of C_4 or



(a) The even process.



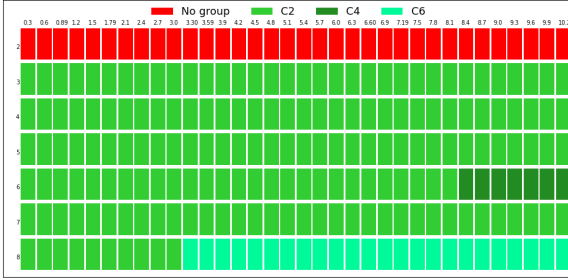
(b) The C_3 cycle.



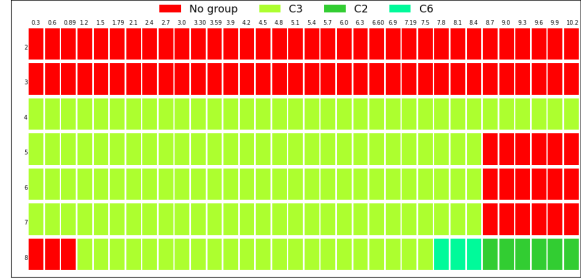
(c) The alternating group.

Figure 2.2: Reconstruction of the probability distribution over strings of length $k = 4$ measured by Hellinger distance. The quality is shown in percentages, where the highest value reflects zero distance between distributions. The left column corresponds to the CSSR algorithm and the right to the n -gram merging method. Every cell of the table corresponds to the algorithm's parameter configuration: the row denotes the history length (from 2 to 8) and the column denotes the significance level (for CSSR) multiplied by 10^2 or the minimization threshold Δ (n -gram merging).

C_6 (that still contains C_2 as a subgroup) was found, showing that excess in splitting the states starts occurring for the higher significance level α .



(a) The C_2 counter.



(b) The C_3 counter.

Figure 2.3: The result of the group reconstruction by CSSR algorithm over the considered parameter range: row denotes the history length (from 2 to 8) and column denotes significance level multiplied by 10^2 . The color reflects the presence of subgroups in automaton’s transformation semigroup: red - aperiodic (no non-trivial subgroups), light green - C_2 , yellow green - C_3 , dark green - C_4 , aqua green - C_6 .

2.7.3 The length three cycle

The next example, shown in Figure 2.4, is chosen to consider if the algorithm can recognize the bigger but still cyclic group C_3 counter. It is just one of the possible constructions where the three states 1, 2, and 3 are permuted by the event A with high probability. The rest of the states and transitions are added to make sure that probability distributions over events are different for all states and no more permutations (like C_2) arise in the transformation semigroup. In this construction, the length of the substrings of consequent A ’s is a multiple of three with high probability, but not always. This makes reconstruction harder than in the previous case, as C_3 counter has less clear impact on vocabulary and has to be recognized from statistics of observations. On the other side, there are some non-occurring words (substrings with neighboring B and C), which is one of the required (but not sufficient) conditions for group reconstruction by the n -gram merging method.

The quality of probability distribution reconstruction is shown in Figure 2.2b. Both algorithms show the same trend as for the even process, with a slight decrease in quality. For CSSR the quality for $n \geq 3$ is between 97.9% and the maximum of 99.05%, with almost no change when varying α . The n -gram merging method also achieves above 98% for $n \geq 3$ and $\Delta \leq 0.17$, with a maximum of 98.9%. However, with higher values of $\Delta \geq 0.17$ there is a noticeably bigger accuracy drop of quality to 88% for $n \leq 6$, 50% for $n = 7$, or even to the lowest 28% for $n = 8$.

The results of reconstructing C_3 group by CSSR are shown in Figure 2.3b. The number of successful instances is a bit smaller than for an even process, however, a clear and wide

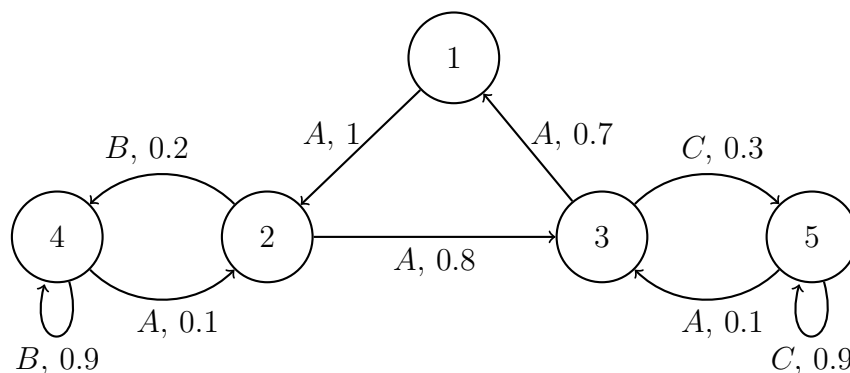


Figure 2.4: The C_3 cycle.

range of successful parameters demonstrates a great ability to recognize C_3 permutation group from the observation sequences in a non-random manner. The n -gram merging method had only three instances of group reconstruction which, however, contained C_2 , but not C_3 group: for $n = 6$ and both $\Delta = 0.065$, $\Delta = 0.075$ and for $n = 7$ and $\Delta = 0.095$. The above is probably a consequence of non-occurring B and C substrings, that allow the method to differentiate states 2 and 3, but not to recognize C_3 permutation by event A .

2.7.4 The alternating group

The previous examples demonstrated the ability of CSSR to recognize the cyclic groups. In a last considered example, shown in Figure 2.5, the reconstruction of a more complex group, non-cyclic group will be considered. It has two generators that act on five states: A permutes two pairs of states (1, 2) and (3, 4) and B is a cyclic permutation of states (1, 3, 5), that give a total of 60 elements. It is called the alternating group A_5 and is the smallest nonabelian simple group. No additional states or transformations were added to the automaton only the transition probabilities were selected to keep the states' distributions different.

The quality of reconstructing the probability distribution by CSSR is above 99% for $n \geq 3$, with the highest of 99.6%, and by the n -gram merging method it also reaches 99.5% for a range of parameters. However, reconstruction quality drops to 64.9% for $5 \leq n \leq 6$ with $\Delta \geq 0.25$ and for $7 \leq n \leq 8$ with $\Delta \geq 0.12$, showing a threshold of collapse of states in merging.

The transformation semigroup of the reconstructed machines is always aperiodic for the n -gram merging method, and almost always for the CSSR method. The two cyclic

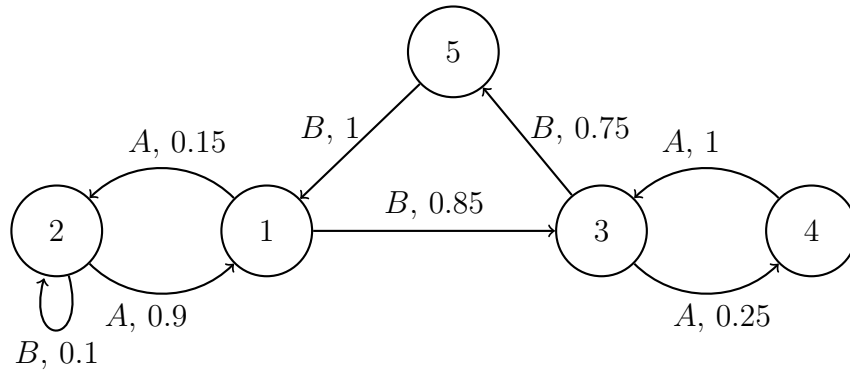


Figure 2.5: The alternating group.

groups C_2 and C_3 were both recognized only by parameters $n = 8$ and $0.084 \leq \alpha \leq 0.102$, but neither of these cases recognized A_5 . It is worth saying that failure to recognize the group might be explained by the low effect of the permutation groups on the statistics or vocabulary of the sequences. That can be emphasized by adding more states and transformations to the system but it is not obvious what type of graph modification will still keep the experiment unbiased and not incorporate too many ‘hints’ for the algorithm.

2.8 Conclusion

This section describes one of the possible algorithms for DPFA reconstruction based on merging the n -gram machine corresponding to the input data. The details of the algorithm are described and its software implementation is created, the similarities of the method with some other reconstruction techniques are briefly mentioned. The two distance metrics, epsilon and Jaccard, are defined to compare the constructed machines. The transformation semigroup of the reconstructed automaton is considered, revealing that it is aperiodic unless the permuted states contain both the n -grams from which permuting transitions are not possible and n -grams from which they are. This condition shows that group reconstruction is more likely the consequence of the imperfect merging, than the property of the method, and suggests that similar state-merging techniques might also have a low potential for group reconstruction.

The examples of the even process, the cyclic group C_3 and the alternating group A_5 were considered for both, the described n -gram merging algorithm (using own implementation) and CSSR method (using the available open source code). For the considered parameters, both methods achieve a high-quality reconstruction of the probability distribution over

strings of length $k = 4$ over a certain parameter range. For CSSR such a great result is obtained with all instances of $n \geq 3$, showing almost no sensitivity to history length or significance level. However, for the n -gram merging method, with the increase of the minimization threshold, the quality eventually drops due to the excessive merging of states, and sometimes it appears at pretty small values, e.g., $\Delta = 0.17$. The ability of the CSSR method to recognize the permutation of states was demonstrated for cycles of small lengths C_2 and C_3 , while a more complicated alternating group was not preserved (for the selected transition probabilities of the automaton).

The presented study does not concentrate on performance improvement or such theoretical analysis as time complexity or error bounds. This would certainly be a goal if the algorithm were considered as a primary independent subject of study, however in this work the algorithm is mainly meant to be the core instrument of the second part of the thesis on using finite automata for analysis of neuroscientific data. This is why we prefer a simpler model, our own implementation, and why we limit analysis of the algorithm to experimental evaluation.

The next chapter considers the application of the finite automaton reconstruction method to distinguish the sequences of EEG microstates. While for both measures, the statistics and permutation structures, better performance was obtained for CSSR (within the considered parameter range), only the n -gram merging method is used further. In the considered application, for example, when two groups are compared, some microstate transition sequences (n grams) are inherent for one but not possible to occur in the other group. As the data is filtered from noise, such words are unlikely to be an error but are probably the markers of the corresponding groups. However, they may be rare in data and represent a transient state, so they will be removed from the automaton by the CSSR method, but will remain as some causal state member in the n -gram merging method. The detection of such patterns is more relevant in terms of considered application than the reconstruction of probability distribution over strings, which however still does serve as a general accuracy measure. Also, CSSR showed low sensitivity to both parameters, history length n and significance level α . While this is good evidence of the stability of the algorithm and the resilience of the reconstructed machine, it suggests the method will be also insensitive to the rare n -gram markers. Additionally, some previous attempts on the CSSR application showed that it may encounter some implementation errors, for example, in the presence of dead states that only occur at the end of lines. The above comments explain some reasoning for using the n -gram merging method in the next chapter. While the utilization of the CSSR method in the same analysis remains a natural future step, that can potentially even demonstrate stronger distinguishability power, it is not conducted within the scope of this thesis.

Chapter 3

Application of Finite Automaton to Distinguish Sequences of EEG Microstates

3.1 Introduction

Electroencephalography (EEG) microstates are topographic maps of the momentary distributions in electric potential that remain quasi-stable in a millisecond time range (approx. 80–150 ms on average) and are separated by rapid spatial configuration changes. They can be obtained from event-related or resting-state data and further analyzed into topological classes and their parameters. The parameters include duration (the average length of time a microstate remains stable), occurrence (the average number of times per second the microstate is dominant during the total recording time), coverage (the fraction of total recording time that the microstate is dominant), and the transition probabilities between microstates. Most resting-state EEG studies (e.g., [36, 33, 40]) reveal four topographic microstate classes, labeled by A,B,C,D and referred to as ‘canonical’ EEG microstates in the literature. These classes account for 70–80% of the variance in topographic structure. The analysis of the standard parameters allows differentiation between sequences of different conditions or tasks e.g., different age groups [37], sleep stages [10], clinical symptoms e.g., schizophrenia [60, 33], or functional significance [1].

Moreover, regularities and patterning in the temporal sequence of EEG microstates, i.e., syntax, have been of growing interest. Particular patterns in microstate syntax may also serve as markers for the important aspects of brain activity. First, [40] showed there is a dif-

ference in the predominance of transitions between certain microstate pairs and sequences of transitions forming a cycle (i.e., $A \rightarrow D \rightarrow C \rightarrow A$) in healthy controls and schizophrenia patients. Further studies involving analysis of the microstate syntax considered differences in transitions in relation to mental disorders, such as schizophrenia and Alzheimer’s disease [48, 41], personality differences [55] or mental processes, e.g., shifting from self-related to self-unrelated task [9], etc. These studies found that the differences in the frequencies of particular transition pairs, in conditional transition probabilities, or in the predominance of a longer transition sequence over its reverse are significant in differentiating the corresponding conditions.

The efficacy of pairwise microstate transitions in reflecting differences between conditions (i.e., for distinguishing them) gave rise to the question of whether they carry enough information to characterize the dynamics of EEG microstate sequence data. Gärtner *et al.* in [26] showed that a reduced first-order Markov chain¹ describes the microstate transitions for resting state EEG in wakefulness and sleep with high accuracy (with only about 2% of mass difference between distributions). Based on this representation, they propose the sampled marked intervals (SMI) model that explains the sequence of EEG microstate at global field power (GFP) peaks as produced by the sparse sampling² from an unobserved background process³. The observation that state duration inferred by SMI is shorter than the known in literature microstate duration was commented in [35] as being due to inappropriate model assumptions (i.e., independence of GFP peak occurrence and topographic change). In addition, [28] pointed out that such a model does not account for observed long-range dependencies in microstate sequences. They demonstrated that self-similarity and long-memory properties are evidently the properties of EEG microstate sequences, which are absent in sequences generated according to estimated Markov transition probabilities. Other evidence showing the need to go beyond pairwise transitions was given in [45]. They considered the length 4 cycles (e.g., ADCA) which were significant for detecting schizophrenic patients in [40] study, and showed the discrepancy between empirically observed and analytically estimated (using pairwise transition model) percentages of these words.

Given the evidence of such longer temporal dependencies, approaches using pairwise transition probabilities may be not sufficient to capture properties of EEG microstate dy-

¹In their model $P[X_i = a]$ depends on whether there was a change of microstate, i.e., $X_{i-1} \neq X_i$, but not on the value of X_{i-1} itself. While in zero-order model, the probability $P[X_i = a]$ has no dependence on X_j , $\forall j < i$ and in the first-order model $P[X_i = a]$ depends on the value of X_{i-1} . Thus, it is the reduction of the first order but more than a zero-order model.

²Where the sampling process is a stationary renewal process with Gamma distributed intervals.

³Assuming the background process can be described as a Poisson process with independent and exponentially distributed intervals, which are marked independently with fixed probability of every mark.

namics, thus a number of studies test for the Markov properties and periodicity of the microstate sequences, adding to the argument for looking beyond first-order microstate transitions. The oscillatory behavior of EEG microstates appears to be inherited from periodicities of global field power (evidently related to the underlying EEG signal frequency), as shown by information-theoretic analysis in [69] and frequency analysis in [70]. In addition, none of the studied resting state EEG microstate sequences fulfills the zero, first, or second-order Markov property in [69], meaning that the memory effect in the sequences extends more than two time steps into the past. However, they show that microstate sequences are short-range correlated for large time lags, meaning that the majority of temporal dependencies in the long run have finite memory. Similarly to [69], the zero and first-order Markov property was rejected for all subjects for EEG sequences in wakefulness and non-rapid eye movement sleep stages in [70], also showing that the memory effect extends for least two sampling intervals. However, in the deeper sleep stage the dynamics was simplified, and for 50% of subjects could be described already by the second-order Markov process, giving evidence of the finiteness of the memory (bounded dependencies). Assessing stationarity of the sequences, [69] found that for the time windows of 10s, 20s and 40s stationarity hypotheses were rejected for 80%, 60% and 44% of subjects, respectively. That demonstrates that empirical microstate transition matrices may frequently lack stationarity, which however appears to increase when evaluated over longer time windows.

There are methods that go beyond pairwise transitions, such as [62] and [3], and analyze transition behavior without relying on Markov properties or stationarity of microstate sequences, as those properties are often not encountered in data. In [62] Tait *et al.* suggested a classification model based on complexity measure and theta relative power combined in a support vector machine classifier. The complexity measure is based on counts of the number of unique sub-sequences within a sequence, which is lower when a small number of patterns are frequently repeated. For the prediction of Alzheimer’s disease (AD), the model had a classification rate of 90.9%, revealing the lower complexity in AD. Artoni and colleagues [3] introduced the Microsynt method for microstate sequence analysis: it creates the representation of words occurring in sequences as a dictionary⁴, where the word length is fixed and is inferred from current data size and complexity⁵. The words from the dictionary are then classified into classes based on their entropy (higher entropy means more “randomness”, lower—more repeated patterns) and the statistical significance (against surrogates) of recurring words in the sequence over time is analyzed. They suggest that Microsynt can be used to reliably differentiate two or more cognitive conditions, showing

⁴The dictionary consists of all occurring n-grams and its representation contains the number of occurrences of each word in data.

⁵The optimal word’s length is the maximum value for which the size of the random and real dictionaries are the same.

experimental study for ‘fully awake’ and ‘fully unconscious’ conditions, and observing the entropy change when progressing between them. The vocabulary extracted by ‘syntax signature’⁶ can provide further insights on transitions, most different in differentiated groups, e.g., revealing different microstate loops in sequences.

In this chapter, we explore using epsilon-machine models to characterize EEG microstate sequence syntax dynamically, i.e., using generative models that go beyond pairwise transition and word structure. The application of this model to microstate sequences allows (i) to account for temporal dependencies longer than pairwise transitions, and (ii) to extract classes of histories based on their transition similarities for further investigation of microstate syntax within causal states. Moreover, (iii) it yields the dynamical system, represented by the probabilistic automaton that allows analysis of its structural and statistical properties, as well as has predictive power, i.e., allows generation of sequences with similar structure and assignment of likelihoods to the sequences. Therefore, this framework rigorously combines the approach of estimating the transition probabilities with the idea of identifying words into classes in a way that extends the framework of data representation into the inference of a dynamical structure accounting for the system’s behavior, allowing the application of additional analysis techniques such as entropy rate, probabilistic grammar, system’s complexity, etc. The idea of using epsilon-machine reconstruction for analysis of EEG microstate sequences was first introduced in [45], demonstrating the reliability of the method and its power in revealing the differences in the underlying system’s structure that are not reflected in the matrix of conditional transition probabilities. In this study, epsilon-machines reconstructed by the algorithm, described in the previous chapter, are applied to distinguish between participant groups (meditators and controls) and cognitive modes (verbalization, visualization, and mind-wandering).

3.2 Methods

The data used in this study was collected as part of another research project at Institute of Psychiatry, Psychology and Neuroscience, King’s College London, supported by the BIAL foundation and a USAFOSR grant with the corresponding scientific reports [2] and [46]. Thus all the stages of cleaning recordings from noise, extracting the microstate maps from the data, and converting the continuous EEG recordings into discrete sequences of EEG microstates were completed at the data collection stage by others, prior to the study conducted within the scope of this thesis. The experimental paradigm, participants, and

⁶The words that best differentiate two conditions are those, which account for 10% of absolute differences in sequence representations.

EEG data processing are described in detail in [1, 2] and briefly outlined in Sections 3.2.1 and 3.2.2. For the provided EEG microstate sequences, the study presented in this thesis addresses the question of distinguishing them by applying the finite automaton model. Thus, for this application, Section 3.2.3 on the new methodology describes the developed analysis framework.

3.2.1 Participants and experimental paradigm

The data used in the study was collected from twenty meditators (Age: mean=48.4, SD=11.1, range: 32-68; Male/Female=13/7) and twenty meditation-naive healthy controls (Age: mean age=40.2, SD=14.63, range: 21-62); Male/Female=12/8). The inclusion criteria for meditators was a regular mindfulness meditation practice (regular practice was defined as at least 40 minutes of formal meditation practice a day at least 5 days a week for at least 2 years). The inclusion criteria for all participants were English fluency and age between 18 and 65 years. The exclusion criteria were a history of i) mental health problems; ii) drug and/or alcohol abuse; and iii) neurodevelopmental and/or neurodegenerative disorders. The study received ethics approval from King’s College London Ethics Approval Board (Ref: HR-16/17-4092) and informed consent was gained from the participants prior to participation.

The experimental paradigm consisted of two repetitions of the same condition sequence: *mind-wandering* → *verbalization* → *visualization*, denoted as *MW1*, *Ver1*, *Vis1* and *MW2*, *Ver2*, *Vis2*. For the *mind-wandering* condition, the participants were instructed to relax and allow their minds to wander naturally. For the *verbalization* and *visualization* conditions, the participants were asked to repeat silently the word ‘square’ or to visualize a square, respectively, at a self-paced rate of approximately 2 sec (without silent counting). Each condition was 3 minutes long, with a total duration of 18 minutes for each run. Participants had their eyes closed throughout the run. All participants were given a practice run of a full condition sequence lasting 1 minute.

3.2.2 EEG microstate computation

EEG data were recorded using a 40-channel Neuroscan Quikcap system (Compumedics, USA): a total of 36 channels were recorded including 4 electrodes for the EOG signal and two references. The data for each participant were segmented into six ~3-min epochs (i.e., 2 repetitions x 3 conditions = 12 epochs), filtered removing the noise and artifacts, and finally, down-sampled to 250 Hz.

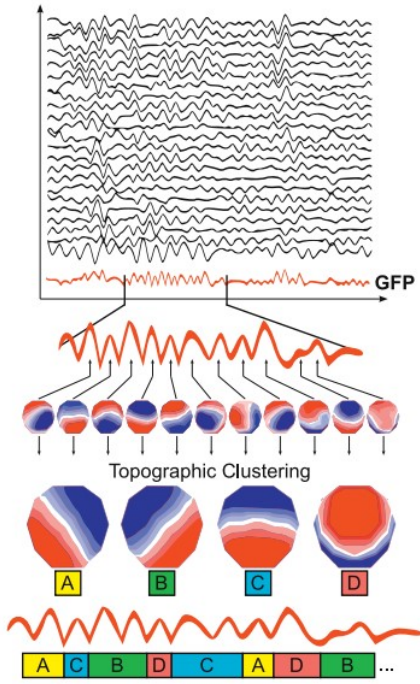


Figure 3.1: The schematic representation of computation of EEG microstate sequences [31].

The general EEG microstate computation procedure, as described in [31] starts by calculating the global field power GFP from the instantaneous potentials V_i across k electrodes as follows:

$$GFP = \sqrt{\left(\sum_i^K (V_i(t) - V_{mean}(t))^2\right)/K}.$$

The GFP signal represents the strength of the electric field over the brain at each instant and is used to describe global brain activity.

The topographic maps at the peaks of the GFP signal are processed by the clustering algorithm, from which a small set of topographic pictures (clusters' centers) are extracted, as schematically shown in Figure 3.1. Remarkably, above 70% of the total topographic variance of EEG recording can be explained by such a small set as four clusters. After extracting the main topographic maps, each of the GFP peaks is assigned the label of the corresponding cluster based on topographic similarity. This results in the discrete sequence of microstate labels. Each of the microstate maps is found to be quasi-stable as it remains dominant for about 80–120 ms before rapidly transitioning to a different map.

For the data used in this study the EEG microstate computation was performed using the EEG Analysis Toolbox KeyPy [42, 43], using the pre-defined microstate classes [43]—four canonical maps labeled by A, B, C, and D shown in Figure 3.2.

The procedure described above gives a microstate sequence calculated from the topographies at the global field power (GFP) peaks, which corresponds to approximately one label for every ≈ 50 millisecond of the recording. This is referred to as a peak-based time scale. Additionally, from each of the recordings for this study, two other microstate sequences were computed, as described above, however for different time scales: clock-based, where the labels describe a sequence of the topographic maps observed approximately every 4 ms, and the event-based time scale, where the microstate labels only denote the change of the dominant microstate, which gives the sequence of distinct microstates corresponding to approximately one letter every $\approx 80 - 120$ ms [45].

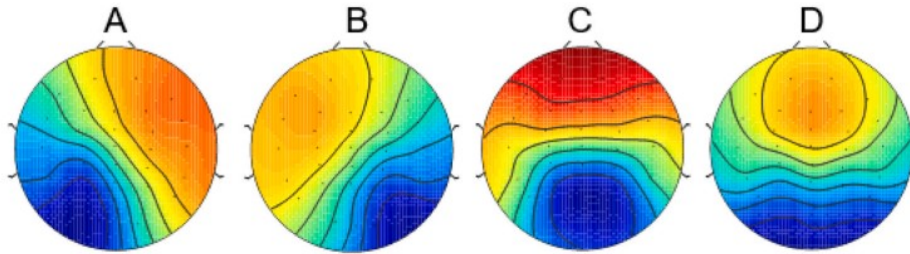


Figure 3.2: The topographical maps for canonical EEG microstate classes A , B , C , and D [43] maps from the KeyPy template library (reduced from 64 to 30 channels and rendered with the EEGLAB topoplot function).

3.2.3 Distinguishing EEG microstates sequences with ϵ -machines

Figure ?? outlines a high-level representation of the methodology of the study. Rather than directly analyzing the EEG microstate sequences this work aims to gain insight on the differences and similarities of the recordings by analyzing the epsilon machines constructed from them. As reflected in Figure ??A instead of using the individual recording, the EEG sequences from all participants in a group are first combined and only then used to derive the automaton. Only recordings that correspond to the same participant's group (M or C), cognitive mode (MW , Ver , Vis), and repetition ($\#1$ or $\#2$) are combined together, which results in twelve group-level data files for each of the three temporal scales (clock, peak, event). The combination of the individual data here means the concatenation of the corresponding EEG microstate sequences, preserving the line breaks. As data is available in three time resolutions and ϵ -machines have two construction parameters: n -gram length and minimization threshold Δ , the search of the optimal model is performed over the range of parameters: $n \in \{1, 2, \dots, 7\}$, $\Delta \in \{0, 0.05, 0.1, \dots, 0.95, 1\}$ for all three temporal scales. Figure ??A gives an illustrative example of the epsilon machine: every node of the graph corresponds to the set of equivalent n -grams of EEG microstates (causal states) and every labeled edge represents a transition with the corresponding event (one of the microstates). Each of the possible transitions from the node has a certain probability of being observed in the model. The color of every node is supposed to reflect the presence of each of the microstates in the n -grams that make up a given vertex of the graph. The complexity of the ϵ -machines derived from the given EEG microstate sequences varies in the considered parameter range from a single-node graph to a graph with over a thousand vertices.

The main method used to address the question of finding similarities and differences in the given EEG microstate sequences is *separability* of the corresponding group-level ϵ -

machines Figure ??B. Both separation between- and within- the groups refer to checking certain separability criteria for two sets of the ϵ -machines. Naturally, in the setting called *between the groups* two sets being separated are the set of all ϵ -machines of the Controls' group and the set of all ϵ -machines of the Meditators' group. In the setting called *within the groups* the separation of ϵ -machines of a particular cognitive mode (MW , Ver or Vis) from the rest of the machines of the same participant's group (M or C) is considered.

The above concept puts the question of differentiating the given EEG microstate sequences into the framework of checking separability in one of the seven possible settings. Both of the two types of separation criteria are based on the distance metrics defined on ϵ -machines - Section 2.5. Conceptual differences between these two criteria is that they capture different type of layout of ϵ -machines in metric space: linear separability Figure ??B-I allows the clusters of machines to be sparse and close to each other, requiring only no overlap, while strict separability Figure ??B-II requires the clusters of the machines to be at least as far from each other as their diameters. Another difference is that strict criterion is applied without dimensional reduction and in the same metric space which is originally defined for ϵ -machines, while linear separability criterion requires the embedding of the original metric space into Euclidean space and then dimensional reduction, as this criterion is applied to a two-dimensional projection of corresponding machines. Euclidean embedding doesn't always preserve all the information, however, the selected projection necessarily captures two dimensions that explain the most variance. For more details about the separation criteria address Sections 3.2.4 and 3.2.5.

The epsilon machines of corresponding EEG microstate sequences are considered over the range of parameters, and the primal question that is answered by those methods is the separability of groups and cognitive modes according to both criteria by assigning the 'yes/no' separability status for every configuration of parameters. Before making any further conclusions about such results the natural question of its reliability arises. It is addressed by the method of detecting false positives over the same parameter range - Sections 3.2.6.

If a relatively small percentage of parameters receives positive separability status, then assessing the percentage of those where the separation criterion holds for the incorrectly partitioned machines, allows us to get an insight into the degree of randomness in the observed separability phenomena.

Potentially, the strict separability criterion captures the stronger differences in the temporal patterns in the EEG microstate sequences than linear criteria. The best cases where the strict separation criterion holds, i.e., those, corresponding to the highest separation ratio, are considered in more detail, as is shown in Figure ??C.

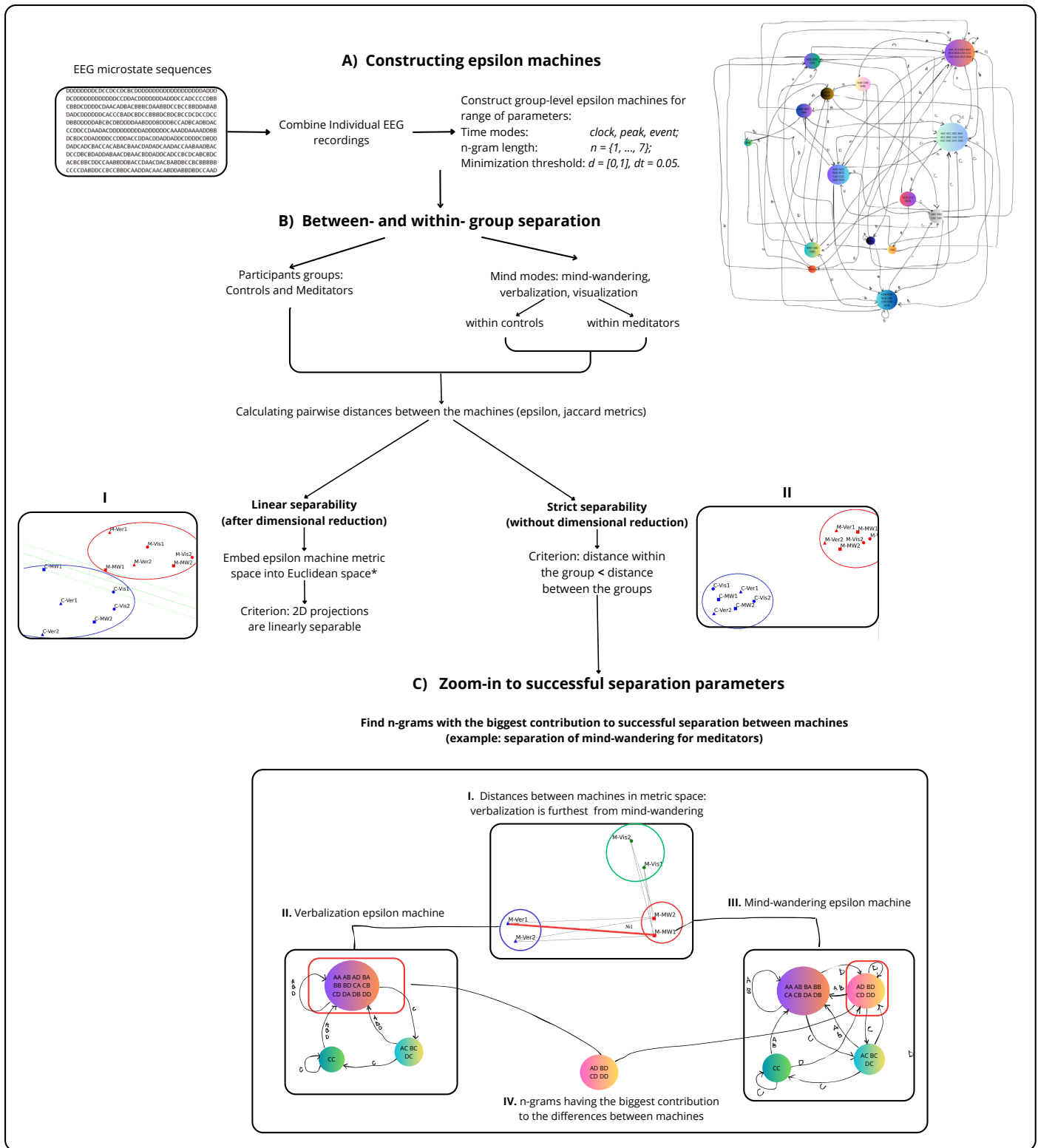


Figure 3.3: Work pipeline, explaining how EEG microstate sequences of two participant groups and three mind modes are analyzed and compared. A) - Group-level ϵ -machines are constructed from combined individual recordings for a range of parameters. B)- For a set of ϵ -machines the question of the group separability and mind-mode separability within the groups is addressed, using two types of criteria: B-I) Linear separability in 2D projection of the Euclidean embedding of the distance metrics; and B-II) strict separability criteria applied to original distance metrics. These criteria capture different types of group-level machine layouts relative to each other in metric space. C) - Successful separations of groups and mind modes for different parameter configurations are explained with the sets of n -grams having the biggest contribution to distance between sets.

The main idea of zooming in on cases of successful separation is to detect the n -grams, that have the biggest contribution to the distances between sets of separated machines, and therefore are distinguishable in particular settings. For the method of detecting such n -grams see Section 3.2.7. Those n -grams are assumed to represent the special temporal patterns in the EEG microstate sequences, that account for the bigger weight in the found separations than other n -grams, and they are further analyzed to explain similarities or differences of corresponding recordings.

The illustration of the above step is given in Figure ??C and corresponds to the within-group separation setting for separating the cognitive mode of mind-wandering from visualization and verbalization in the Meditators' group. The corresponding parameter settings are peak-based temporal scale and $n = 2, \Delta = 0.15$ machine construction parameters. Figure ??C gives a schematic picture of how ϵ -machines are located in a metric space, highlighting the distances from both mind-wandering machines to four other machines of the Meditators' group from which they are being separated. The distance between the pair of ϵ -machines of mind-wandering $MW1$ and verbalization $Ver1$ is highlighted with a red line because this is a pairwise distance from which the block $I^* = \{AD, BD, CD, DD\}$ comes - the block of n -grams with the biggest contribution to separating sets of ϵ -machines. Figure ??C-II, III shows graphs of corresponding epsilon machines. The block I^* is the intersection of causal states and it makes the only difference in the partition of n -grams in the set of causal states for these two machines. It creates a separate causal state in mind-wandering but is part of the bigger causal state in the verbalization machine. The way the block I^* is detected is by comparing the transition probabilities from causal states which are omitted in the figure.

3.2.4 Linear separability of sets of epsilon machines after dimensional reduction

The concept of linear separability of the sets of machines relies on the defined epsilon and Jaccard distances, however, it is not directly applied to ϵ -machines in the corresponding metric space, but to the corresponding real vectors in the Euclidean space. Although separability can be potentially verified in any dimension \mathbb{R}^n , this study focuses on the embedded vectors after dimensional reduction to \mathbb{R}^2 . For two sets of ϵ -machines \mathcal{A} and \mathcal{B} the criterion of *linear separability* requires the sets of corresponding real vectors $X^{\mathcal{A}} \in \mathbb{R}^2$ and $X^{\mathcal{B}} \in \mathbb{R}^2$ to be linearly separable. The sets of points $X^{\mathcal{A}}$ and $X^{\mathcal{B}}$ in \mathbb{R}^2 are two-dimensional projections of the embedding of sets of ϵ -machines \mathcal{A} and \mathcal{B} into Euclidean space.

Embedding the Metric Space in Euclidean Space

Let \mathcal{A} be a set of ϵ -machines and $D \in \mathbb{R}^{n \times n}$ denote the distance matrix in one of the defined metrics $d : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$ (epsilon or Jaccard), i.e., D_{ij} is the distance between epsilon machines $\mathcal{E}_i \in \mathcal{A}$ and $\mathcal{E}_j \in \mathcal{A}$. The algorithm in [71] suggests the following. Define matrix M as

$$M_{ij} = \frac{D_{1j}^2 + D_{i1}^2 - D_{ij}^2}{2}, \quad M = V\Lambda V^{-1},$$

where Λ, V denotes eigenvalue decomposition of matrix M : Λ - diagonal matrix of eigenvalues, V - normalized matrix of eigenvectors. Then Euclidean embedding $f : (\mathcal{A}, d) \rightarrow \mathbb{R}^{n \times n}$ can be computed as

$$f(\mathcal{A}, d) = X = Re(V\sqrt{\Lambda}),$$

where $\sqrt{\Lambda}$ denotes the matrix with singular values in the diagonal. Every i -th row of the resulting matrix X describes the coordinate of the i -th embedded ϵ -machine, columns correspond to the dimensions of Euclidean space. If there exists an embedding that preserves all information into the lower-dimensional space $k < n$ then all columns corresponding to the extra dimensions will be zero in X . The final step of constructing $X^{\mathcal{A}}, X^{\mathcal{B}}$ is to select the two dimensions that explain the most variance, for the above procedure they correspond to the two biggest eigenvalues of M . Corresponding columns of X give the projections the separation criterion focuses on:

$$\forall \mathcal{E}_i \in \mathcal{A}, X_i^{\mathcal{A}} = [X_{i,1}, X_{i,2}] \text{ where } X = f(\mathcal{A}, d), \quad \Lambda_{1,1} \geq \dots \geq \Lambda_{n,n} - \text{eigenvalues of } M,$$

To assess the quality of displaying real distances between ϵ -machines in \mathbb{R}^2 , the following numerical characteristics are calculated: maximum local distortion, maximum global distortion, variance, maximum Euclidean embedding error.

3.2.5 Strict separability of sets of epsilon machines without dimensional reduction

For the set of ϵ -machines \mathcal{A} to be *separable from* set \mathcal{B} according to the strict separation criteria the distance between any two pairs of machines $\mathcal{A}_1, \mathcal{A}_2 \in \mathcal{A}$ from set \mathcal{A} is required to be less than the distance from any machine $\mathcal{A} \in \mathcal{A}$ from set \mathcal{A} to any machine $\mathcal{B} \in \mathcal{B}$ from set \mathcal{B} :

$$\max_{\mathcal{A}_1, \mathcal{A}_2} (d(\mathcal{A}_1, \mathcal{A}_2)) \leq \min_{\mathcal{A}, \mathcal{B}} (d(\mathcal{A}, \mathcal{B})),$$

where $d(\mathcal{E}_1, \mathcal{E}_2)$ denotes the distance between two epsilon machines \mathcal{E}_1 and \mathcal{E}_2 . The alternative formulation is that set \mathcal{A} is strictly separable from set \mathcal{B} or the strict separation criteria holds for the set \mathcal{A} compared to set \mathcal{B} . The above criterion can be applied to both epsilon and Jaccard distance metrics, indicating the separability of the sets of machines in the corresponding metric space. Conceptually such a criterion requires the set of ϵ -machines to be at least as far from any machine of the other set as its diameter. The above may hold for either both or only one of the compared sets of machines, which implies two types of separability.

Comparing two sets \mathcal{A} and \mathcal{B} of ϵ -machines one is said to have *no strict separation between \mathcal{A} and \mathcal{B}* if strict separation criterion doesn't hold for any of the compared sets, *one-sided separation of set \mathcal{A} from set \mathcal{B}* if the criterion holds for set \mathcal{A} , but not for the set \mathcal{B} , and *two-sided separation of \mathcal{A} and \mathcal{B}* if the criterion holds for both sets. One-sided separation is particularly preferred in the within-group separation setting, as separating the set \mathcal{A} of one mind mode from the set \mathcal{B} of the rest in this participant's group, the set \mathcal{B} can have a big diameter as other modes may be far from each other as well.

In the presence of such a separation, a numerical measure of it is introduced as follows. The *separation ratio* $R_S(\mathcal{A}, \mathcal{B})$ for two sets of machines \mathcal{A} and \mathcal{B} is defined as the ratio of the minimum distance between machines from different sets and the diameter of the union of these sets:

$$R_S(\mathcal{A}, \mathcal{B}) = \frac{\min_{\mathcal{A}, \mathcal{B}}(D(\mathcal{A}, \mathcal{B}))}{\max_{\mathcal{C}, \mathcal{C}'}(D(\mathcal{C}, \mathcal{C}'))}, \quad \mathcal{A} \in \mathcal{A}, \quad \mathcal{B} \in \mathcal{B}, \quad \mathcal{C}, \mathcal{C}' \in \mathcal{C} = \mathcal{A} \cup \mathcal{B}.$$

The idea that motivates such a measure can be explained as follows. Suppose one considers two sets of points in \mathbb{R}^2 that are linearly separable with strong decision boundaries. The ratio of the separation margin to the diameter of the union of two sets can approximately explain what fraction of the area of the convex hull of those points falls within the separation margin. Therefore higher value of the separation ratio R_S indicates better clustering of the separated sets of machines.

Since the separation criterion requires inequality between maximum and minimum distances, meeting this criterion guarantees that inequality holds for all pairs of distances: any distance between machines from the same set does not exceed the distance between machines of different sets. Although in case the criterion is not satisfied, it doesn't tell how many pairs of distances violate the desired inequality. Suppose we are comparing two sets of machines: \mathcal{A} , which has n machines, and \mathcal{B} , which has m machines. There are $\binom{n}{2} + \binom{m}{2}$ pair of distances that correspond to machines from the same set and $n \cdot m$ corresponding

to machines from different sets. Denote the total number of required inequalities as T :

$$T = \left[\binom{n}{2} + \binom{m}{2} \right] \cdot n \cdot m.$$

For every pair of machines $\mathcal{A}_1, \mathcal{A}_2 \in \mathcal{A}$ from the set \mathcal{A} define the set $V(\mathcal{A}_1, \mathcal{A}_2)$ of all pairs of machines from different sets that are at least as far from each other as $\mathcal{A}_1, \mathcal{A}_2$ are:

$$\forall \mathcal{A}_1, \mathcal{A}_2 \in \mathcal{A} \quad \text{denote} \quad V(\mathcal{A}_1, \mathcal{A}_2) = \{(\mathcal{A}, \mathcal{B}) \in \mathcal{A} \times \mathcal{B} \mid d(\mathcal{A}, \mathcal{B}) \geq d(\mathcal{A}_1, \mathcal{A}_2)\}.$$

Define the *percentage of separation* $P_S(\mathcal{A}, \mathcal{B})$ of two sets of ϵ -machines \mathcal{A} and \mathcal{B} as the percentage of pairs of distances within these machines that satisfy the inequality corresponding to the strict separation criteria:

$$P_S(\mathcal{A}, \mathcal{B}) = \left(\sum_{\mathcal{A}_1, \mathcal{A}_2 \in \mathcal{A}} |V(\mathcal{A}_1, \mathcal{A}_2)| + \sum_{\mathcal{B}_1, \mathcal{B}_2 \in \mathcal{B}} |V(\mathcal{B}_1, \mathcal{B}_2)| \right) \cdot \frac{100\%}{T}.$$

This measure indicates how close are two compared sets of ϵ -machines to satisfy the separation criterion. The averaged over the range of parameters separation percentage tells how likely it is for these sets of ϵ -machines to be separated with a strict criterion.

3.2.6 Methods validation using permuted data

For every configuration of parameters, both separation criteria give a boolean verdict on separability, to assess the reliability of this verdict perform the validation with the permuted data. There are 12 group-level machines in total, 6 in every group. Consider every possible way of dividing epsilon machines into two groups of equal size, excluding the partition based on the real data. Calculating the proportion $\beta \in [0, 1]$ of the permutations that are separable, obtain the probability of random permutation to satisfy the separation criterion.

To assess the reliability of results with the separation percentage P_S compare distributions and statistics (mean, median, mode, standard deviation) for the separation percentage of observed data with those permuted over the range of parameters.

3.2.7 Identifying n -grams with the biggest contribution to separation of sets of epsilon machines

For every fixed pair of machines $\mathcal{E}_1 = (S_1, A, \delta_1, p_1)$ and $\mathcal{E}_2 = (S_2, A, \delta_2, p_2)$ constructed for history length n every n -gram is associated with a certain state and every causal state

is a set of n -grams. Consider partitioning all n -gram that occur in at least one machine into the set of blocks - the intersections of causal states denoted as $\{I_{\mathcal{E}_1, \mathcal{E}_2}\}$. Both epsilon and Jaccard metrics consist of the summation of the terms over all possible n -grams in the alphabet A , such that all n -grams that come from the intersection of the causal states of two machines have the same corresponding terms and contribute to the distances with equal quantities. Define the functions for epsilon and Jaccard metrics that for every intersection of the causal states of epsilon machines $\mathcal{E}_1, \mathcal{E}_2$ assigns its *weighted contribution* to distance between those machines:

$$\forall \mathcal{E}_1, \mathcal{E}_2 \quad C_E, C_J : \{I_{\mathcal{E}_1, \mathcal{E}_2}\} \rightarrow \mathbb{R}, \quad \forall I \in \{I_{\mathcal{E}_1, \mathcal{E}_2}\}, \quad \text{such that } I = s_1 \cap s_2, \quad \text{for } s_1 \in S_1, s_2 \in S_2,$$

$$C_E(I) = |I| \cdot \left[\sum_{a \in A} |p_1(s_1, a) - p_2(s_2, a)| \right],$$

$$C_J(I) = |I| \cdot \left[1 - \frac{\sum_{a_i \in A} \min(n_1([w]_1 s_1, a_i), n_2(s_2, a_i))}{\sum_{a_i \in A} \max(n_1(s_1, a_i), n_2(s_2, a_i))} \right].$$

Selecting key n -grams for distinguishing between the machines or the sets of machines

For the fixed pair of machines \mathcal{E}_1 and \mathcal{E}_2 consider the total ordering \geq_E (and similarly ordering \geq_J) of the blocks of n -grams $\{I_{\mathcal{E}_1, \mathcal{E}_2}\}$ to be defined as:

$$\forall I_1, I_2 \in \{I_{\mathcal{E}_1, \mathcal{E}_2}\} \quad I_1 \geq I_2 \quad \Leftrightarrow \quad C_E(I_1) \geq C_E(I_2),$$

meaning that the blocks of n -grams are listed in decreasing order of their importance in distinguishing between \mathcal{E}_1 and \mathcal{E}_2 . Denote the ordered set of blocks as $\{I_{\mathcal{E}_1, \mathcal{E}_2}\}_{\geq_E}$ and note that the sum of their contributions equals the distance between machines $d_E(\mathcal{E}_1, \mathcal{E}_2)$. Then for every fixed threshold $\gamma \in (0, 1)$ one can select the most important n -grams that account for γ fraction of the distance between machines $I_{\mathcal{E}_1, \mathcal{E}_2}^*$ as the union of the heaviest blocks:

$$I_{\mathcal{E}_1, \mathcal{E}_2}^* = \bigcup_{i=1}^k I_i, \quad I_i \in \{I_{\mathcal{E}_1, \mathcal{E}_2}\}_{\geq_E}, \quad k = \arg \min_j \left\{ \sum_{i=1}^j C_E(I_i) \geq \gamma \cdot d_E(\mathcal{E}_1, \mathcal{E}_2) \right\}.$$

Alternatively, only the first block I_1 from the ordered set $\{I_{\mathcal{E}_1, \mathcal{E}_2}\}_{\geq_E}$ can be considered to explain the differences in distinguishing between corresponding machines. This gives us two ways of selecting the key n -grams when comparing epsilon machines.

Consider the sets of ϵ -machines \mathcal{A} and \mathcal{B} . Denote the disjoint union of the causal state intersections for all pairs of machines from considered sets:

$$\{I_{\mathcal{A},\mathcal{B}}\} = \bigsqcup_{A \in \mathcal{A}, B \in \mathcal{B}} \{I_{A,B}\}.$$

Applying the total ordering \geq_E (or \geq_J) and two methods of identifying distinguishing n -grams for the set $\{I_{\mathcal{A},\mathcal{B}}\}$ naturally generalizes the above concept from comparing pair of machines $\mathcal{E}_1, \mathcal{E}_2$ to comparing two sets of machines \mathcal{A} and \mathcal{B} . Note that the above union is disjoint as even in the case when the intersection of causal states for different pairs of machines are the same, their weighted contribution to the distances may be different, as it relies on the transition probabilities in the corresponding machines.

3.3 Results

The standard way of comparing sequences by analyzing such microstate parameters as duration, occurrence, and coverage, was conducted for the provided data [2] outside the scope of this work. However, it did not reveal any significant differences between groups or conditions enough to differentiate them or to be recognized as markers of corresponding brain activity. The detailed results of differentiating the EEG sequences achieved by separating epsilon-machines are reported below.

3.3.1 Separating participant groups: meditators and cotrols

Linear separability

Figure 3.4 shows the successful separation cases over the considered range of parameters, using epsilon and Jaccard distance matrices respectively. In 827 out of 882 cases considered configurations (93.8%) participant groups are separable with linear criterion. A similar test for n -gram machines gives a positive verdict in 41 out of 42 cases (93.7%).

Figure 3.5 illustrates ϵ -machines embedded into the Euclidean space in the 2D projection for three configurations of parameters. The upper panel corresponds to the n -gram machines, and the lower one shows how the projections change after minimization. The selected parameters of minimized machines correspond to cases where participant groups are separated with both criteria, and strict criterion gives the highest separation ratio: 0.935, 0.940, and 0.887. Comparing 2D projections of ϵ -machines embedded into Euclidean

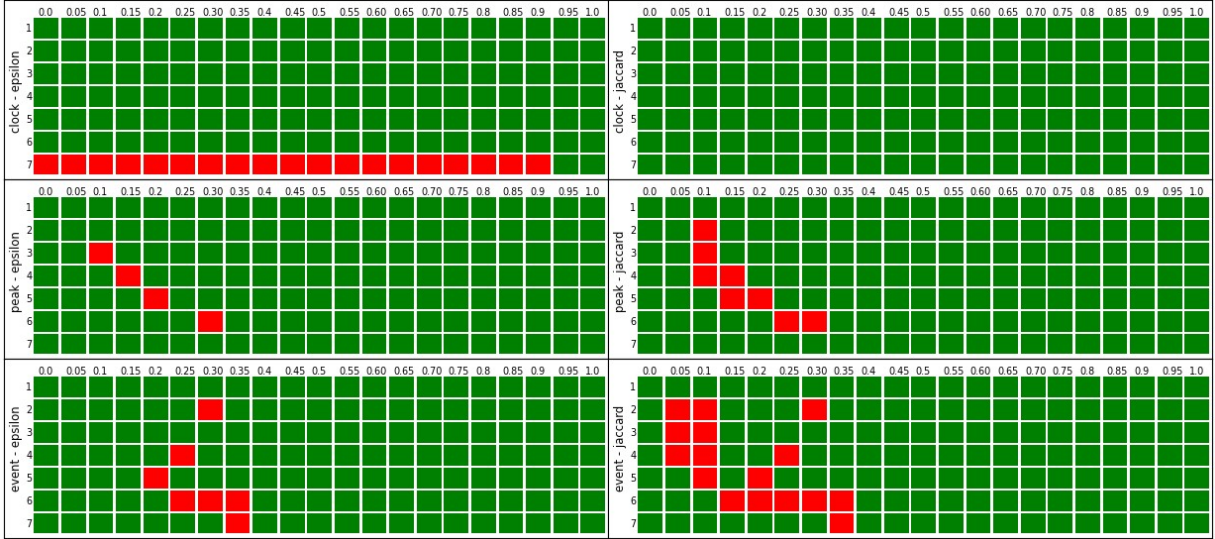


Figure 3.4: The result of differentiating participant groups: meditators and non-meditators, using linear separation criteria using epsilon (left) and Jaccard (right) distance metrics for three temporal scales (clock, peak event). The green and red colors reflect the success and failure of separating group-level epsilon machines respectively.

space, observe that the points corresponding to minimized machines are grouped better, decision boundaries are stronger and the separation bandwidth is closer to the diameter of the set of points. The variance of the selected dimensions for every panel of Figure 3.5 is at least 82% – 96% for n -gram machines and 91% – 98% for minimized machines. The average variance of the selected dimensions over all configurations of the parameters equals $V = 85.14\%$ with standard deviation $\sigma_V = 7.59$, which shows that in general selected display dimensions represent a high percentage of information, but have high dispersion as well. The lowest and highest variance are $V_{min} = 59.28\%$ and $V_{max} = 99.95\%$.

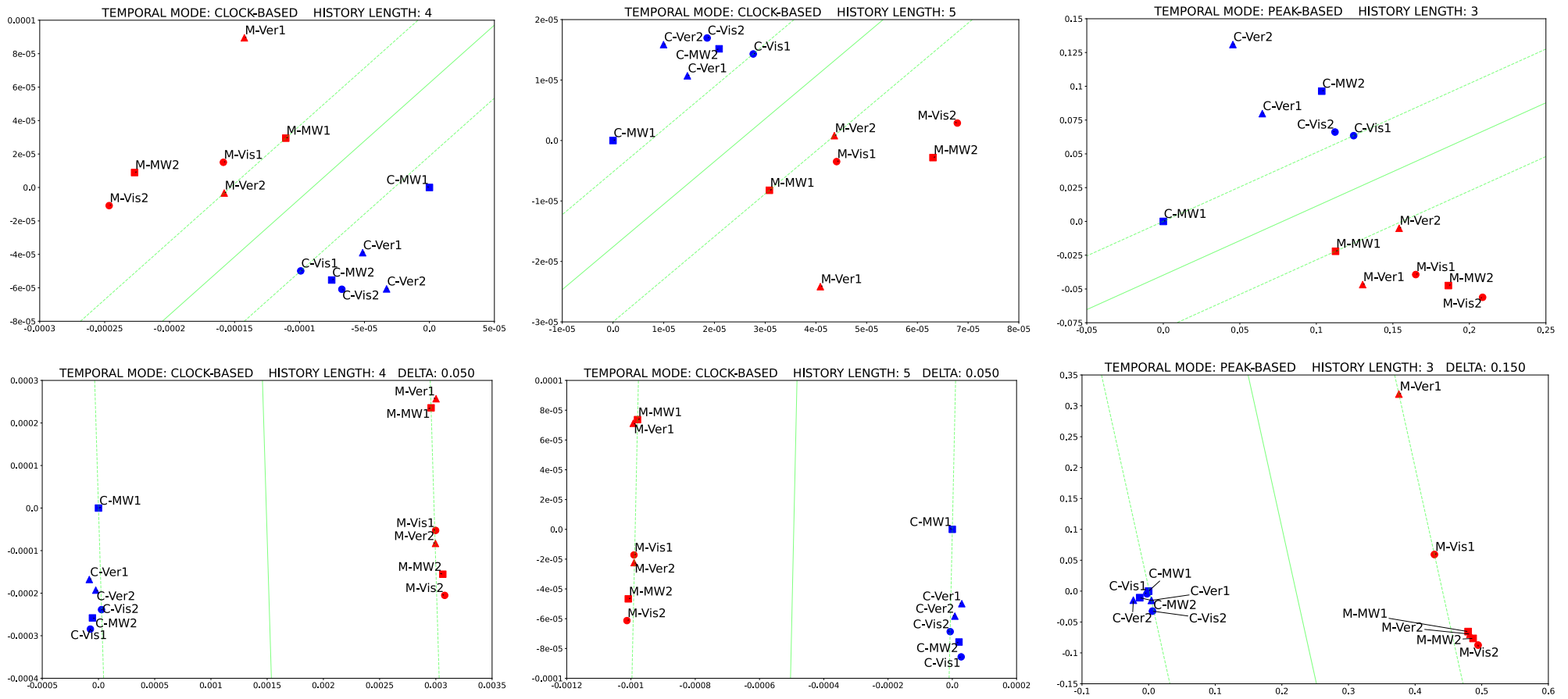


Figure 3.5: 2D projection of Euclidean embedding for n -gram machines (first line of figures) and minimized machines (second line) for selected cases of two-sided separation by strict criterion. The first and second columns represent the case of clock time mode, using epsilon distance type, where minimization is done with $\delta = 0.05$, history length $L = 4$ (first column), and history length $L = 5$ (second column). The third column represents the case of peak time mode, using jaccard distance variation, history length $L = 3$, where minimization is done with $\delta = 0.15$.

Strict separability

Table 3.1 lists the parameters for which the strict separation criterion is met and provides corresponding separation ratio R_S values. The largest separation ratio for two-sided separation equals $R_S = 0.940372$ and was observed for the following parameters: clock time mode, history length $n = 5$, minimization threshold $\Delta = 0.05$, and epsilon distance metric. For one-sided separation of controls the largest separation ratio equals $R_S = 0.44061$, of meditators $R_S = 0.97323$.

Table 3.1: Successful cases of strict separation between meditators and non-meditators.

Type	Time scale	n	Δ	Metric	R_S
two-sided $C \leftrightarrow M$	clock	4	0.05	epsilon	0.94
		5			
		4		Jaccard	0.68
		5			
peak	3	0.15		0.88	
one-sided $C \rightarrow M$	peak	3	0.15	epsilon	0.44
		1	0.2	Jaccard	0.14
one-sided $C \leftarrow M$	peak	6	0.15	epsilon	0.89
			0.2		0.87
		7	0.0	Jaccard	0.97
			0.05		
			0.1		
			0.2		

The above information shows that the group of controls is clustered and separable more often than the one of meditators and, for parameters corresponding to separation cases, has more than twice higher separation ratio.

Validation using permuted data

To validate the separations between the groups of meditators and non-meditators as described in Section 3.2.6 consider 462 possible partitions of twelve group-level ϵ -machines in two groups, one of which is correct for each of the 882 parameter configurations.

Linear separation criterion. The probability of having a linear separation for randomly permuted data is at most $\beta = 0.02$ for any of the 882 parameter configurations. Observed

data are separable with linear criterion in 93.88% of parameter configurations, while for all possible permutations of data, the percentage of parameters that gives linear separation has mean $\mu = 1.35$, range $[0; 42.97]$ and standard deviation $\sigma = 4.97$. Those conclude that cases of successful separations with linear criterion shown in Figure 3.4 are reliable.

Strict separation criterion. For each of the 882 parameter configurations, the result of the permutation test shows that $\beta = 0$ i.e., none of the 461 incorrect partitions of the set of machines into groups have a two-sided separation with strict separation criterion, which allows the conclusion that cases of successful two-sided separation in Table 3.1 are reliable.

Separation percentage. The separation percentage P_S defined in Section 3.2.5 measures how close the sets of epsilon machines are to be separated with a strict criterion. That is, P_S shows how well two sets are clustered as opposed to being widespread and close or even having the overlap.

Figure 3.6 demonstrates the distribution of two-sided separation percentage between two participants' groups for observed and permuted data over the considered range of parameters. The clear difference is that distribution of observed data is stick to the right, most of the time giving values above 70 – 75%, while distribution of the permuted data stick to the left, falling between 40 – 50% for most of the parameters. The permuted data behaves noticeably the same in all time scales, however performances of observed data are almost indistinguishable in clock- and peak- based time scales, while event clearly doesn't reach high values of 90 – 100% in a single case.

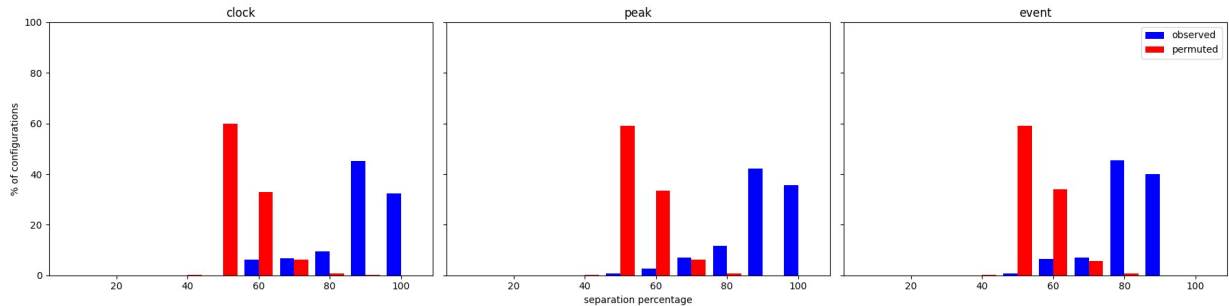


Figure 3.6: The distribution of separation percentage between meditators and non-meditators comparing observed and permuted data.

Table 3.2 summaries statistical information on two-sided separation percentage illustrated in Figure 3.6. As can be seen, the separation of observed data outperforms the permuted in every presented measure. The clear difference is that permuted data has an

average percentage just below 50%, as well as the median and mode in all three time scales. At contrast separation percentage for observed data is above 83% on average in clock and peak time scales and slightly below 80% for event. Similarly, median and mode, being close to the mean follow the same trends.

Table 3.2: Statistical measures for two-sided separation percentage between groups.

Type	Time scale	Mean	St. dev.	Median	Mode	Range
Observed	clock	83.12	9.79	85.93	85	[54.8 , 100.0]
	peak	84.07	9.90	84.03	85	[46.1 , 100.0]
	event	78.21	8.99	77.59	75	[47.9 , 89.10]
Permuted	clock	49.92	5.78	48.89	45	[36.5 , 90.0]
	peak	49.92	5.81	48.80	45	[32.0 , 90.3]
	event	49.93	5.79	48.61	45	[33.7 , 97.7]

Figure 3.7 outlines how the separation percentages of the observed data discussed above are located in the grid of considered parameter configurations. In common, there is the range of fluctuations around the minimization threshold $\Delta \in [0.05, 0.3]$ and overall difference in performance of epsilon (on the left) and Jaccard (on the right) distance metrics: epsilon distance outperforms Jaccard in clock-based time scale, but the opposite is true for event- and peak- based time scales. For $\Delta > 0.4$, the separation percentage between groups reaches a stable plateau, which is the same for all n -gram lengths and does not change with further minimization. This stability does not necessarily imply an absence of changes in machines and distances; it could be that the separation percentage is not sensitive enough to detect these changes. Further method development is required for an insight into the structure of causal states in order to make the method more sensitive to subtle changes in ϵ -machines.

n -grams with the biggest contribution to separation

Focusing on the two-sided separation cases enumerated in Table 3.1 select the ones with the highest separation ratio R_S for each of the time scales. Following the procedure described in Section 3.2.7 the ϵ -machines of separated sets are compared to extract the sets of n -grams, coming from the ϵ -machines' causal state intersections, that have higher contribution to the distance and account for some percent of the observed separability. The details about the parameters of the selected separation cases and the extracted sets of n -grams is summarized in Table 3.3. In all selected separation cases between and within both groups, the intersection happens to be equal to one of the causal states, however it

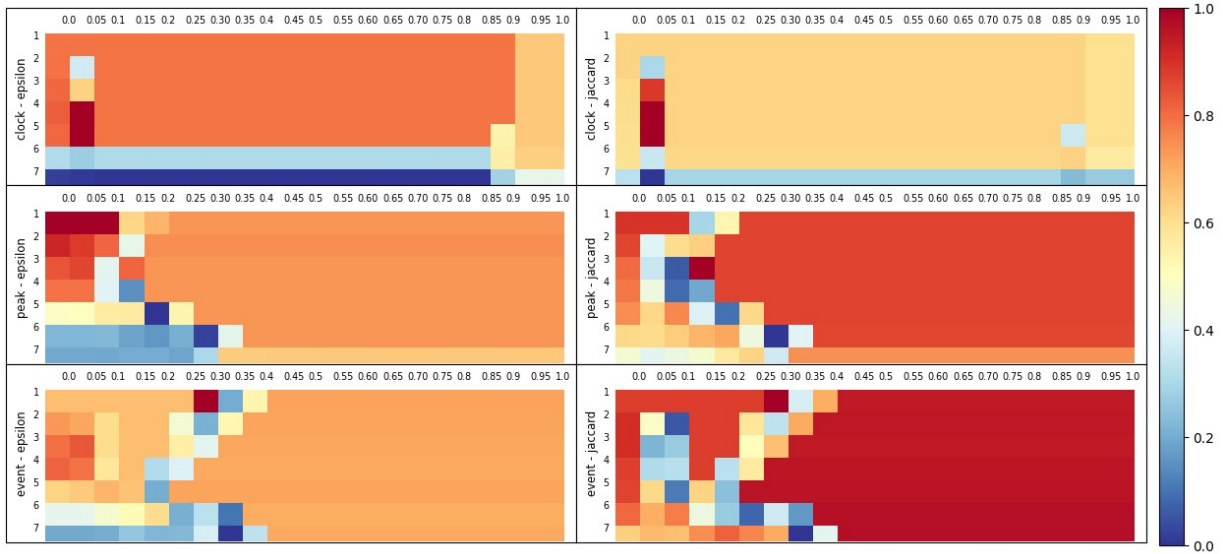


Figure 3.7: The separation percentage between meditators and non-meditators over parameter range.

is not necessarily the case in general. A regular expression describing all n -gram in the intersection I_1 is marked by the \star symbol next to the corresponding causal state. The three last columns of Table 3.3 give information about the union of the heaviest blocks I^* that sums to at least 10% of the distance for the two selected separation cases: number of blocks in the union, the total number of n -grams there and regular expression describing I^* for the cases where the compact representation was found ⁷.

For clock mode $I^* = (A^+|B^+|C^+)DD^+ \cap (A|B|C|D)^5$ is the union of the 15 blocks containing 9 distinct n -grams coming from different pairs of the ϵ -machines. The union I^* accounts for 10.02% of the distance and consists of the five grams ending with at least two D microstates, but not starting with D . The n -grams with the longer suffix of D have a bigger contribution to the distance between groups. The first block I_1 comes from the pair of machines C-Vis1 and M-Vis2, and the rest of the blocks in the union I^* come from the controls' visualization machine in pair with meditators' machines of different modes.

For peak mode $I^* = I_1 = (A|B|C|D)^2(A|B|C)$ is the union of 6 blocks containing 48 distinct n -grams that account for 10.0% of the distance between groups. These are all three-grams not ending with D . The first block I_1 comes from the pair of machines C-Ver2

⁷The regular expressions given in the table were found with the help of undergraduate research assistant Valerie Fernandez.

Table 3.3: Selected cases of strict separation with highest separation ratio (i) between control (C) and meditator (M) groups, (ii) between a cognitive mode (mind-wandering (MW), verbalization (Ver), visualization (Vis)) and the other two modes within groups. Parameters are abbreviated for *time scale*: clock c, peak p and event e; values following ‘n’ and ‘d’ correspond to the *n*-gram length and *minimization threshold* Δ ; *distance metric*: last letter j stands for Jaccard and e for epsilon. I_1 denotes the block of *n*-grams (intersection of causal states) having largest contribution to distance between separated sets of ϵ -machines; its regular expression is marked by \star in the ‘Causal State’ column (in the selected cases, this intersection always coincides with one of the causal states). I^* denotes the union of the blocks with largest contributions to distance accounting for at least 10% of distance. A dot ‘.’ denotes any letter, successive dots can be different letters. \dagger can never have two of the same letters appear consecutively. A dash ‘-’ replaces regular expressions that are too long.

Type	Parameters	Separation Ratio R_S	Distance % due to I_1 (I^*)	ϵ -machines of block I_1	Causal States (Regular Expression, restricted to words of length n) for I_1	Blocks in I^*	n -grams in I^* (distinct)	Union I^* ($\geq 10\%$ distance) (Regular Expression)
C-M	c-n5d0.05-e	0.94	0.67 (10.02)	C-Vis1	$(A^* B^* C^* D^*)D^+$	15	45 (9)	$(A^+ B^+ C^+)DD^+$
				M-Vis2	$\star(A B C)DDDD$			
	p-n3d0.15-j	0.89	1.74 (10.00)	M-Vis2	$(A B C D)^3$	6	288 (48)	$(A B C D)^2(A B C)$
				C-Ver2	$\star(A B C D)^2(A B C)$			
M-MW	p-n2d0.10-j	0.29	8.99(17.78)	M-Ver1	$(A B C D)(A B D)$	2	16 (8)	$(A B C D)(A B)$
				M-MW1	$\star(A B C D)(A B)$			
	e-n7d0.30-j	0.80	0.27(10.02)	M-Vis2	-	56	2056 (1084)	-
				M-MW1	$\star(A B C D)^3(A B)DAD - .CBA\dots\dagger$			
M-Ver	p-n2d0.10-e	0.74	8.8(16.78)	M-Ver1	$(A B C D)(A B D)$	2	16 (8)	$(A B C D)(A B)$
				M-Vis2	$\star(A B C D)(A B)$			
	e-n5d0.05-j	0.87	0.17(10.0)	M-Ver2	$(A B C D)(A C)DAD^\dagger$	115	172 (91)	-
				M-Vis2	$\star(A B D)CDAD$			
M-Vis	c-n6d0.90-e	0.19	14.6(14.6)	M-Ver2	$(A^* B^* C^* D^*)(A^+ B^+)$	1	32 (32)	$(A^* B^* C^* D^*)(A^+ B^+)$
				M-Vis2	$\star(A^* B^* C^* D^*)(A^+ B^+)$			
	p-n3d0.05-j	0.75	0.73(10.46)	M-MW1	$(A B C D)DA$	19	25 (12)	$(A B C D)DA (A B D)BC $ $ (A B)(AD CB) ABD$
				M-Vis2	$\star ADA BDA$			
	e-n7d0.05-e	0.94	0.018(10.0)	M-Ver1	$BDCBACB$	902	908 (605)	-
M-Vis2				$\star BDCBACB$				
C-MW	e-n2d0.05-j	0.54	3.7(10.91)	C-Vis2	$AC BC$	3	3 (2)	$AC BC$
				C-MW1	$\star AC$			
C-Ver	c-n6d0.90-e	0.15	11.76(11.76)	C-MW1	$(A^* B^* C^* D^*)(A^+ B^+) ACCCCA$	1	32 (32)	$(A^* B^* C^* D^*)(A^+ B^+)$
				C-Ver1	$\star(A^* B^* C^* D^*)(A^+ B^+)$			
	p-n3d0.05-e	0.77	0.73(10.22)	C-Ver2	$(A B C)CA$	19	38 (22)	-
				C-Vis2	$\star ACA BCA$			
	e-n7d0.20-j	0.98	0.056(10.0)	C-Ver1	$(A B C D)^2(A B C)DCDC - .AB\dots\dagger$	600	2132 (864)	-
C-Vis2				$\star(A B C D)(B C D)ADCDC^\dagger$				
C-Vis	c-n2d0.00-j	0.42	2.48(11.15)	C-Ver2	AB	5	5 (3)	$AB BA BC$
				C-Vis2	$\star AB$			
	p-n7d0.15-j	0.99	0.003(10.0)	C-Vis2	$(A B C D)(A C D)DDDDD$	9089	10124 (3481)	-
				C-Ver1	$\star(A B C D)ADDDDD$			
	e-n7d0.10-j	0.97	0.02(10.0)	C-Ver2	$(A B C D)(B C)DCDCD^\dagger$	1318	1570 (939)	-
C-Vis2				$\star(A C D)BDCDCD$				

and M-Vis2, and the rest of the blocks in the union I^* come from machines of controls' mind-wandering and verbalization in pair with meditators' machines of different modes.

3.3.2 Separating cognitive modes in meditators

Linear separability

The linear separation criterion is satisfied for only one cognitive mode, but not two other in 0.79% of parameter configurations for mind-wandering (MW), 4.64% - for verbalization (Ver) and 15.5% - for visualization (Vis). Two of the mind modes are linearly separable with the third one located between separation lines in the 2D projection in 0.45% of parameter configurations for MW and Ver, 1.02% - for MW and Vis and 1.24% - for Ver and Vis. All three cognitive modes are linearly separable in 1.24% of parameter configurations.

Figure 3.8A illustrates for which parameter configurations the linear separation in meditators group is observed. It can be clearly seen in the table, that linear separation of visualization is prevalent in the region corresponding to the epsilon distance metric and clock time scale for $n \in [1, 5]$ and $\Delta \in [0, 0.9]$, and is noticeably present for event time scale, $n \in [1, 3]$ and $\Delta \in [0.1, 0.25]$. As opposed to epsilon distance metric, Jaccard one seems to be completely unsuccessful in separating any of the modes within meditators in clock time scale. The linear separation of verbalization and mind-wandering appears to be rare phenomenon, indicating the substantial overlap between these two mind modes, preventing the separation. To have non-zero chances of linear separability of at least one mind mode it is suggested to prioritize the choices of parameters corresponding to lower left corner of the table, namely $n \in [3, 7]$, $\Delta \in [0, 0.3]$.

Strict separability

Table 3.4 lists all cases where one-sided separation of epsilon machines is found for cognitive condition within meditators group. Out of 882 parameter settings there are only 2 (0.23%) separations of mind-wandering, 3 (0.34%) of verbalization and 6 (0.68%) of visualization. The largest separation ratio $R_S = 0.94$ corresponds to the separation of visualization in event time scale, $n = 7$ and $\Delta = 0.05$, using epsilon distance metric. Generally, the separation ratio is higher for longer n -gram length, however, for the listed separation cases the event time scale gives noticeably higher values of separation ratio R_S than clock and peak.

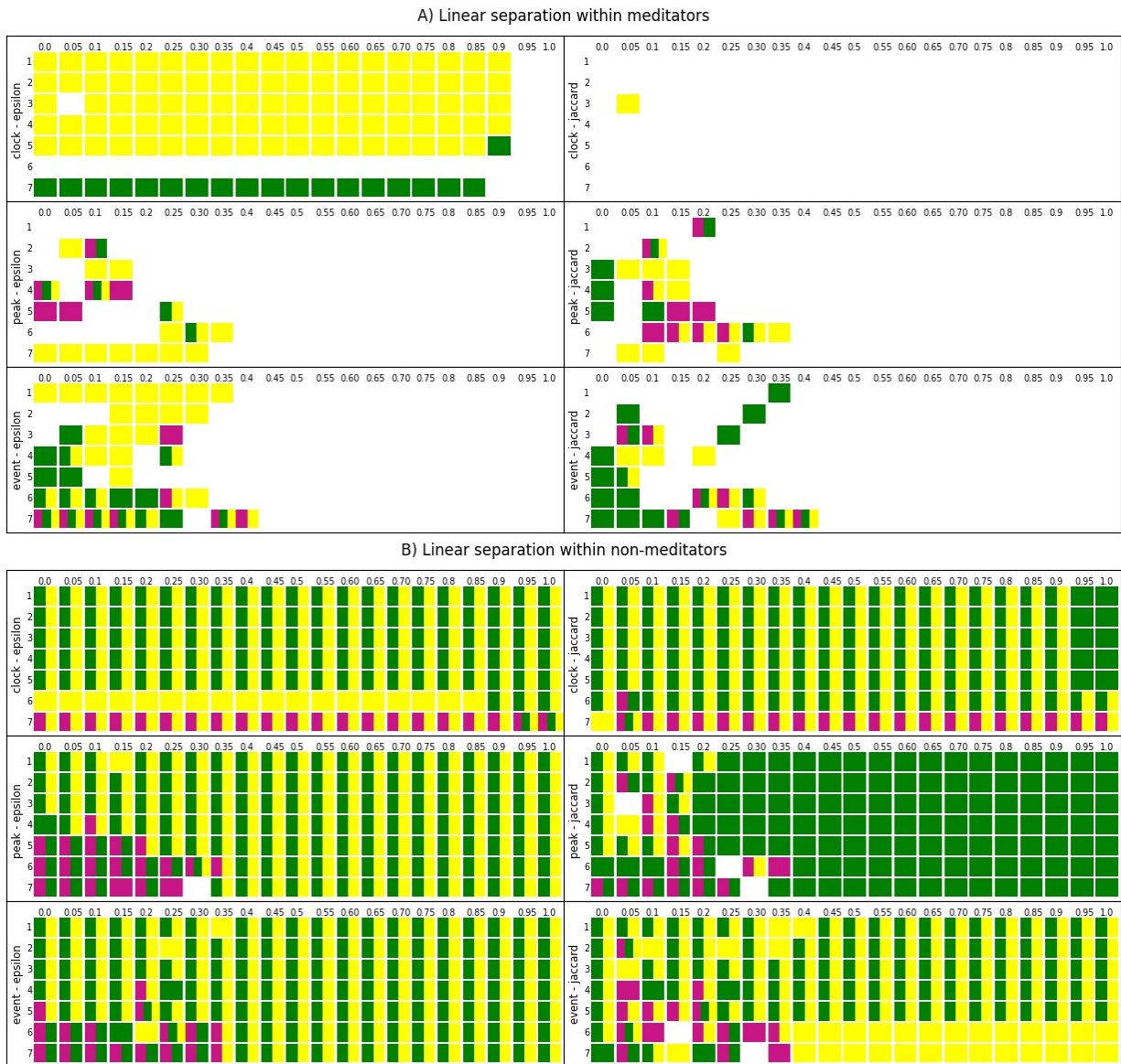


Figure 3.8: The success of linear separation of cognitive modes within meditators - panel A, and non-meditators - panel B, using 6 group-level machines over parameter range. Colors within each cell denote linear separability of mind mode: purple - mind-wandering, green-verbalization, yellow-visualization.

Table 3.4: One-sided separation of cognitive conditions within the meditators group.

Mode	Time	n	Δ	Metric	R_S
MW	peak	2	0.10	Jaccard	0.29
	event	7	0.30		0.80
Ver	peak	2	0.10	epsilon	0.74
	event	5	0.05	Jaccard	0.70
Vis	clock	6	0.90	epsilon	0.19
	event	7	0.05		0.94
			0.15		0.93
	peak	2	0.05	Jaccard	0.55
		3			0.75
		4			0.61

Separation percentage

Consider the separation percentage which is the measure of clustering that indicates how close the ϵ -machines' sets are to be separable with a strict criterion. Figure 3.9 demonstrates the distribution of one-sided separation percentage of cognitive modes within meditators' groups for observed and permuted data over the considered range of parameters. The clear difference is that the distribution of permuted data seems to be uniform, while the one of observed data is primarily located in the lower half of the range, most of the time giving values between 20% and 40%.

The average separation percentage values for mind-wandering, verbalization, and visualization respectively are 15.64%, 25.46%, and 49.10% using clock time scale; 16.07%, 33.37%, and 37.58% for peak time scale; 20.40%, 36.13% and 43.06% for event time scale. That is separability (clustering) improves from mind-wandering to verbalization to visualization using either of the time scales. The mode values of separation percentage for each cognitive mode fall in the same range for clock, peak, and event time scales: within [10 – 20]% for mind-wandering, which is much lower than for verbalization and visualization which are within [30 – 40]%. None of the cognitive modes has a higher average of the separation percentage than randomly permuted data, which reaches a 55% in all time scales.

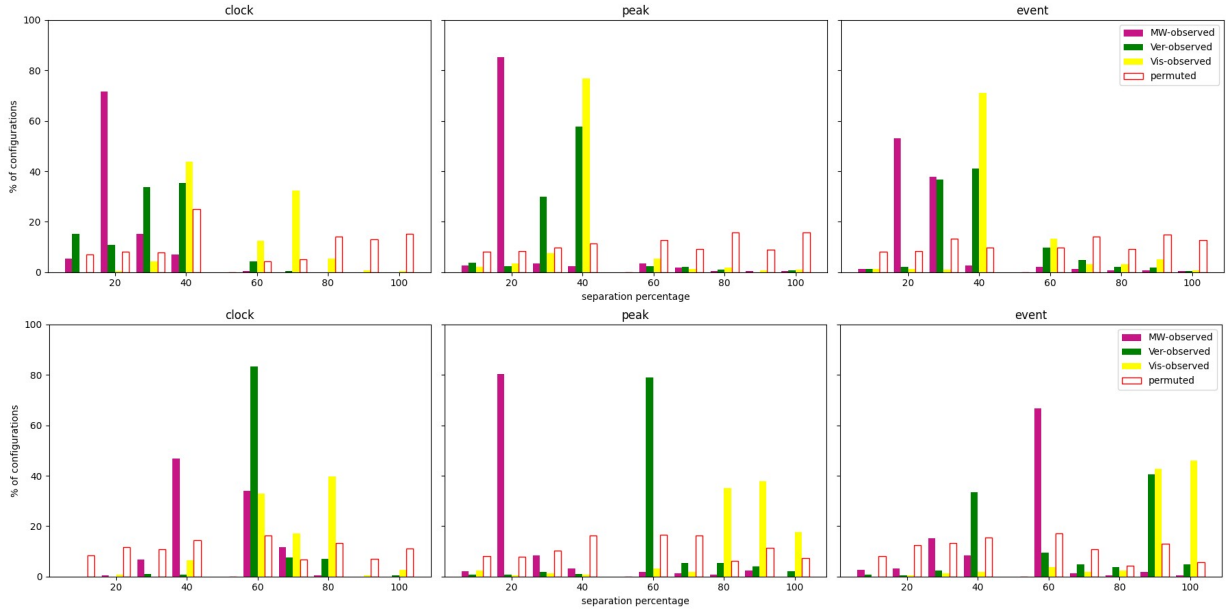


Figure 3.9: The distribution of separation percentage of cognitive modes within meditators - panel A, and within non-meditators - panel B, comparing observed and permuted data.

n-grams with biggest contribution to separation

The details about the extracted sets of *n*-grams having higher importance for observed separations are given in Table 3.3. In the separation of mind-wandering and verbalization in peak the first two blocks I_1 and I_2 are the same and consist of the 8 two-grams ending with *A* or *B* that give 17.18% of the distance from mind-wandering to other modes and 16.78% of distance from verbalization to the other modes. For the event time scale, again in both separation of mind-wandering and verbalization, the seven-grams or five-grams with suffix *DAD* appear to be in the first block I_1 , however, its weight in the distance between separated sets of ϵ -machines is considerably small - much less than one percent, which does not allow to conclude on their importance/significance in the separation of corresponding mind modes. The union I^* of the heaviest blocks needed to accumulate 10% of the distance includes 56 blocks with 1084 distinct *n*-grams for mind-wandering and 115 blocks with 91 distinct *n*-grams for verbalization, where each of the blocks has almost indistinguishable equal contribution to the distance.

The only cognitive mode separated in meditators using a clock time scale is visualization. The first block I_1 consists of the 34 six-grams ending with at least one *A* or *B* accounting for 14.6% of the distance between visualization and other modes. In contrast,

separating visualization in peak and event time scales the weight of the first block in the distance is much less than one percent, and 19 blocks with 12 distinct 3-grams are needed to account for 10% distance in peak, and 902 blocks with 605 distinct 7-grams for event mode. The regular expression describing the union of blocks for visualization in peak is $(A|B|C|D)DA|(A|B|D)BC|(A|B)(AD|CB)|ABD$ that describes 12 out of 64 possible 3-grams. However, for the other cases where the union of many blocks was needed to account for 10% of distance, the structure of the n -grams does not have many syntactical similarities to allow description by the short regular expression.

3.3.3 Separating cognitive modes in non-meditators

Linear separability

The linear separation criterion is satisfied for only one cognitive mode, but not two other in 0.79% of parameter configurations for mind-wandering (MW), 14.7% - for verbalization (Ver) and 6.46.5% - for visualization (Vis). Two of the mind modes are linearly separable with the third one located between separation lines in the 2D projection in 4.53% of parameter configurations for MW and Ver, 6.23% - for MW and Vis and 65.3% - for Ver and Vis. All three cognitive modes are linearly separable in 1.13% of parameter configurations.

Figure 3.8B illustrates for which parameter configurations the linear separation in non-meditators group is observed. An evident conclusion is that linear separation of both visualization and verbalization with mind-wandering between them is a common phenomenon for $n < 6$, except using peak time scale and jaccard distance metric, where separability of verbalization is still prevalent, but not visualization, which for these parameters appears to be too close to mind-wandering. The mind-wandering itself is rarely separable with linear criterion, however the observed cases suggest a clear trend in parameter settings: $n = 7$ in clock time scale with almost no difference with Δ values, and $n \in [5, 7]$, $\Delta \in [0, 0.3]$ in peak and event time scale, which corresponds to lower left corner of the table. There is almost no difference in the performance of distance metrics when separating this mode.

The performance of mind-wandering within both groups is around the same, except that clock time scale doesn't give separations in meditators, but does in controls, specifically for a high length $n = 7$. The performance of verbalization and visualization is substantially different for two groups of participants, with the key feature being in the impact of minimization. If you select the higher values of Δ which causes a lot of n -grams to be merged into each causal state, the mind modes within meditators become indistinguishable, while

the differences between them within the group of controls still can be captured with linear separation criterion.

Strict separability

Figure 3.10 outlines all successful cases of strict separations of cognitive modes within non-meditators. Out of 882 parameter settings there is only one (0.11%) separation of mind-wandering, 21 (2.38%) of verbalization and 192 (21.77%) of visualization, which is considerably higher than the two other modes. Most of the separation instances of visualization correspond to high degrees of minimization $\Delta > 0.3$ with $n = 7$ in peak time scale and for any n in event time scale.

Separation percentage

Consider the separability measure relaxed from binary separation criteria to real-valued in $[0 - 100]\%$ separation percentage. Figure 3.9B demonstrates the distribution of one-sided separation percentage of cognitive modes within the group of non-meditators over the considered range of parameters. Comparing the two participants' groups, notice that while for cognitive modes within meditators, most of the separation percentage values are in the lower half of the range, but within non-meditators, separation percentage values are higher, and most of them are located in the higher half of the range. However, the distribution of permuted data appears to behave the same as within meditators having uniform distribution.

The average separation percentage values for mind-wandering, verbalization, and visualization respectively are 43.83%, 52.55%, and 62.5% using clock time scale; 17.60%, 53.44%, and 80.22% for peak time scale; 43.66%, 63.86% and 89.15% for event time scale. As for meditators, the separability (clustering) improves from mind-wandering to verbalization to visualization using either of the time scales. For visualization cognitive mode over 90% of parameter configurations give very high values of separation percentage: 70 – 100% in event time scale and 80 – 100% in peak time scale. The separation percentage average for randomly permuted data reaches a bit above 49% for clock and peak and 46% for event time scales, which is lower than averages corresponding to verbalization and visualization cognitive modes but higher than the ones corresponding to mind-wandering.

***n*-grams with biggest contribution to separation**

For the selected separation cases (with the highest separation ratio values) the details of the n -grams most contributing to the separation cases are given in Table 3.3. The highest considered n -gram length $n = 7$ gives almost the maximum possible value of R_S for separation of verbalization in the event (0.98) and visualization machines in peak (0.99) and event (0.97) time scales.

The only observed case of the separation of mind-wandering corresponds to the event time scale using $n = 2$. The first block $I_1 = \{AC\}$ consists of the single two-gram and the union of three blocks that account for 10.99% of the distance between mind wandering and other modes consists of $\{AC, BC\}$. A key role in separation verbalization in non-meditators using a clock time scale is to play the six grams ending with at least one A or B . The first block contains 32 six-grams and accounts for 11.76% of the distance. For the separation of visualization in clock time scale the three two-grams $\{AB, BA, BC\}$ are the union of the first three blocks and account for the 11.15% of the separation distance.

In the rest of the separation cases of Table 3.3 can observe the following patterns: transition DC occurs in the heaviest block I_1 in the event time scale, $n = 7$, when separating verbalization - suffix $ADCDC$, and visualization - suffix $DCDCD$; for the peak scale separating visualization, I_1 consists of four 7-grams ending with A followed by five D s. However, all of the above cases have the very light weight of the first block in the overall distance between separated sets, and the union of many blocks is needed to explain 10% of the separation distance. For verbalization, 19 blocks with 22 3-grams are needed in peak and 600 blocks with 864 7-grams are needed for event time scale. Considerably more n -grams are needed to account for 10% of the distance for separating visualization cognitive: the union of 9089 blocks with 3481 7-grams in peak and the union of 1318 blocks with 939 7-grams needed in event time scales.

3.4 Discussion

The application of finite state automata – epsilon-machines (ϵ -machines) for distinguishing the EEG microstate sequences was studied, considering the data of two groups: meditators and meditation-naive healthy individuals, as well as three cognitive conditions within each group: mind-wandering, verbalization and visualization. The methodology incorporated the separation of sets of ϵ -machines based on the distance in the metric space. The separability was achieved by applying two (binary) criteria: linear and strict, with the degree to

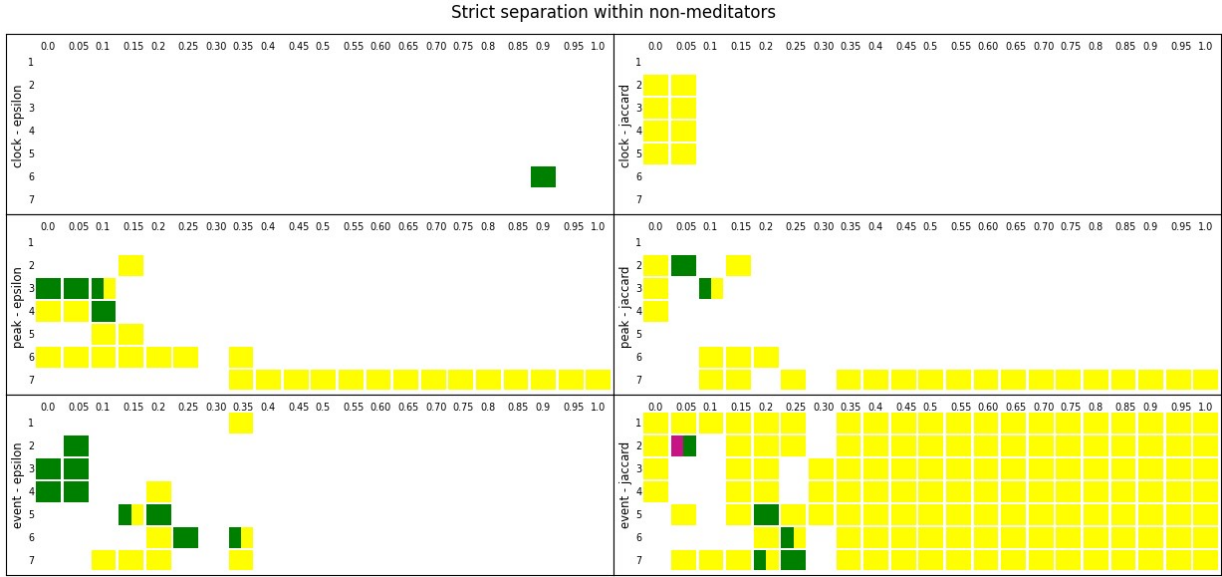


Figure 3.10: The success of strict separation of cognitive modes within non-meditators, using 6 group-level machines over parameter range. Colors within each cell denote linear separability of mind mode: purple - mind-wandering, green-verbalization, yellow-visualization.

which the strict criterion was satisfied indexed by separation percentage (numerical). The methods were also validated using permuted data.

When separating the groups using the observed data, the two groups' ϵ -machines were separable with the linear criterion for 93.88% of considered parameter configurations, as compared with the permuted data, where, on average, only 1.35% of all considered parameter configurations separated the ϵ -machines with permuted group labels (out of all possible 461 permutations). For the strict criterion, there were much fewer cases of separation as compared with the linear criterion (13 and 827, respectively, out of 882 parameter configurations); however, for the strict criterion there were no cases of successful separation using the permuted data. Using the separation percentage as an index of separability (instead of a strict binary 'yes/no' approach) for the observed data, 80% of parameter configurations resulted in separation percentage values above 70%, as opposed to the permuted data where 60% of the parameter configurations resulted in the separation percentage values below 50%.

The separation of ϵ -machines of the two groups (meditators vs. non-meditators) was considerably more robust than for the three cognitive conditions (mind-wandering, verbalization, and visualization) within the groups. The separability of ϵ -machines for the

three conditions was considerably more successful in the non-meditator group than in the meditator group using both the binary separation criteria (linear and strict) and separation percentage, with superior performance for separating ϵ -machines for verbalization and visualization conditions than for mind-wandering in both groups. For the linear criterion, the percentages of parameter configurations that separate a given cognitive condition from the other two for non-meditators and meditators are, respectively: 12.6% and 3.51% for mind-wandering, 85.7% and 7.59% for verbalization, 79.1% and 19.0% for visualization. For the strict criterion, corresponding percentages are: 0.11% and 0.23% for mind-wandering, 2.38% and 0.34% for verbalization, and 21.77% and 0.68% for visualization for non-meditators and meditators, respectively. Overall, the ϵ -machines for the visualization condition in non-meditators were 8.3 times more separable than the other two modes within this group, as well as 17.45 times more separable than the ϵ -machines of the three conditions from each other in meditators. When relaxing the strict criterion to assess separation percentage, consistent separation was only achieved for visualization in non-meditators, with values above 70% in most considered parameter configurations.

In relation to the n -gram lengths, using $n = 1$ for ϵ -machine construction (corresponding to comparing pairwise transition probabilities when using epsilon distance or a number of occurrences of transitions when using Jaccard distance) is as successful as other n -gram lengths for the linear criterion for separating the groups and for verbalization and visualization conditions within the non-meditators. However, for the strict criterion, which requires machines to be clustered, $n = 1$ is not sufficient for successful separation, requiring longer n -gram lengths, particularly $3 \leq n \leq 5$ for two-sided group separation and $2 \leq n \leq 7$ for separating cognitive conditions within the groups. This is a significant finding from applying our method, since the investigation of the EEG microstate sequences of lengths longer than one is beyond the capacity of the traditionally studied transition-probability between EEG microstates (e.g., [43, 1]). This suggests that ϵ -machines are able to capture differentiating aspects of EEG microstate dynamics that are not detectable by the standard EEG microstate parameters (duration, coverage, and occurrence) and pairwise transitions. Indeed, we do not observe significant group differences using these microstate parameters or pairwise transition probabilities for the data in this study.

We have also explored different degrees of minimization (Δ) of the ϵ -machines. With higher degree of minimization, more n -grams are combined into causal states and treated as equivalent in ϵ -machines, rather than being treated separately in terms of their contribution to predicting the next causal state. Moving from coarse to fine-grained, the two extremes are $\Delta = 0$, where transitions from every occurring n -gram matter, and $\Delta = 1$, where only frequencies of transitions for each microstate matter (but not n -grams matter). For the linear criterion, the degree of minimization is not crucial in separating the groups, meaning

they are distinguishable at both extremes and in-between the extremes. However, less minimization seems to be more advantageous when separating cognitive conditions, most likely due to higher minimization resulting in the loss of necessary granularity (detail) to distinguish between the conditions. The strict criterion appears to be even more sensitive to minimization, with threshold $0.05 \leq \Delta \leq 0.3$ being most optimal for separating both groups and conditions. This conclusion on the preferred parameter range for this study is based on the success of the strict criterion, which captures pronounced changes in ϵ -machines' separability.

The (binary) linear separability of the two groups' ϵ -machines constructed using three times scales (clock, peak and event) was comparable; however, the separation percentage (a measure of clustering) was somewhat higher for temporal scales clock and peak (80 – 100%) than for event (70 – 80%) for most parameter configurations. When distinguishing cognitive conditions within the groups, peak (80.22%) and event (89.15%) performed better than clock (62.5%), on average (across all parameter configurations), for separating visualization from the other two conditions in non-meditators. However, the differences in the average percentage separation for the three timescales were relatively small, suggesting that all three timescales were effective for the current data. The relative advantage of using clock rather than event when separating the groups and vice versa when separating the conditions within the groups is likely to be related to the factors (information) that each timescale allows to 'encode' within the ϵ -machines. Thus, clock timescale contains the highest granularity of information on the EEG microstate duration in the 'raw' EEG microstate sequence, less so for peak, and none at all for event. This granularity, however, cannot be retained for ϵ -machine construction using clock timescale, since capturing the total duration of a microstate within one n-gram would typically require $n > 30$, which is computationally infeasible with the current methods. For the short n-gram lengths ($n < 8$) considered in this study, many repetitions of the same 'letter' lead to degeneracy of the captured 'vocabulary' (n -grams occurring in data), which is more severe for clock than either peak or event timescales. Consequently, one of these two factors, information on duration vs. vocabulary richness, has more impact on the structure of ϵ -machines, most likely accounting for the relative differences in separability: duration information in the case of clock time scale, and vocabulary richness in the case of the event timescale. However, for the ϵ -machine constructed using peak timescale, both factors appear to be more balanced. Another observation that can be made is of the relative advantage of epsilon distance over Jaccard distance in clock time scale and vice versa in event time scale. Following the interpretation above, it could be that epsilon distance may be more effective in capturing the impact of duration on distinguishability, while Jaccard could be more suitable where vocabulary richness holds greater significance for distinguishability.

The cases of strict separations between and within the groups corresponding to the highest separation ratio values were considered in more detail to extract the n -grams that can be highlighted in the observed separations of ϵ -machines' sets. Structurally, highlighted n -grams are members of the union of causal states' intersections for ϵ -machine pairs, which account for at least 10% of the distance between separated sets. Comparing ϵ -machines between the groups of meditators and non-meditators in clock time scale the highlighted set of n -grams consists of all five-grams ($n = 5$) ending with at least two D microstates, but not starting with D; here n -grams with the longer suffix of D's have a bigger contribution to the distance between the groups. Complementarily, using the peak-based time scale to separate the groups the highlighted set of n -grams consists of all three-grams ($n = 3$) not ending with D. Separating the cognitive modes within the meditator group, the set of all n -grams ending with either A or B microstate is highlighted for all three cognitive modes: as two-grams ($n = 2$) using peak time scale when separating mind-wandering (accounting for 17.78% of distance) and verbalization (16.78% of distance); and as six-grams ($n = 6$) using clock time scale when separating visualization (accounting for 14.6% of the distance). For cognitive modes within non-meditators, six-grams ($n = 6$) ending with A or B are also highlighted using a clock time scale, but when separating verbalization (11.6% of distance). The shorter length $n = 2$ in selected separation cases within non-meditators highlights the specific words for separation of mind-wandering: AC, BC in event scale, accounting for 10.91% of the distance; and for separation of verbalization: AB, BA, BC in clock scale, accounting for 11.15% of the distance. Recall that $n = 2$ in ϵ -machines differs from traditionally considered pairwise transitions, by corresponding to the distribution of third letter transitions after two-grams. However, for cognitive modes within both groups in around half of the selected separation cases the weight of the causal states intersections in the distance between ϵ -machines was considerably small (much less than 1%). Therefore the set of n -grams accounting for at least 10% of the distance required the union of many causal state intersections (from 19 to 9098 intersections, not having many syntactical similarities), which all have almost indistinguishable contributions to the distance. These separation cases correspond to higher lengths $n = 5$ or $n = 7$ using the event time scale in meditators and event or peak time scale in non-meditators. The above demonstrates that where the separation of cognitive modes within the groups can be detected with ϵ -machines, looking at the transitions from the specific subset of the n -grams is often not enough to account for the differences. Instead, the distinguishable differences between cognitive modes lie in the transition probabilities accumulated among all causal states. In this way, using ϵ -machines allows detecting more separations of cognitive modes than does using traditional parameters or even n -gram frequencies.

The separability of ϵ -machines, both between and within the groups, in future studies might possibly be improved by increasing the length of the EEG microstate sequence. It

was previously reported low short-term test-retest reliability of EEG microstate parameters [1] when comparing four repetitions of the same condition over two successive runs in healthy individuals (with some sample overlap with the present study) and suggested that future studies should collect EEG data by repeating the same conditions in shorter runs (e.g., a break after each repetition), which might potentially improve test-retest reliability across the repetitions of the same condition by reducing participant’s fatigue that results in the alpha power drop affecting the overall reliability of EEG microstates. Therefore, the recommendation for applying the methods reported here is to increase the EEG microstate sequence data used for epsilon-machine construction by concatenating the EEG microstate sequences from multiple repetitions of the same condition presented to the participants with a break in between each repetition to improve the discriminant ability and reliability of the method. Future research should also determine optimal EEG recording lengths (EEG microstate sequence length) required for separability of the ϵ -machines in different contexts (e.g., clinical groups, cognitive processes, etc.) for different sets of parameters.

Although the test-retest reliability of EEG microstates in the current sample was not investigated, it is possible that the lack of significant differences in the standard EEG parameters (duration, occurrence, coverage, and transition probabilities) of the three cognitive conditions between- and within- groups is due to the within-subject parameter variations between the two repetitions of each condition. Nevertheless, the differences are detectable using ϵ -machines modeling the syntax at the ‘vocabulary’-level dynamics of EEG microstate sequences with a broad range of parameter settings (n -gram length, minimization threshold Δ) and all temporal resolutions (clock, peak, event). It was previously demonstrated that the spatiotemporal neurodynamics characterized by EEG microstate sequences require a richer model to capture its complex syntax [45], with the results of the present investigation yielding further support for using methods sensitive to higher-level grammatical structure. Epsilon-machines yield new insight and allow for exploration of the syntactic structure of dynamic systems generating EEG microstate sequences arising in different cognitive states and in different populations.

In order to combine data from multiple participants it is necessary to give labels to momentary maps to classify each pattern of electrical activation as a standard microstate following a well-defined method. Here, the closest Milz maps with standard ”alphabet” A,B,C,D [43] was used. Although the explained variance of microstate topography is lower than for data-driven maps, by skipping the step of finding group-level maps, this approach provides a computationally quick approach to EEG sequence generation that could be readily applied to construct epsilon-machines in close to real-time for future applications. The standard maps used in this study already showed separability of machines, however future research should compare the template map labelling with data-driven microstates in

the same sample in terms of separation success using epsilon-machines, which should only get better with additional tuning and customization, but this was outside of the scope of the current study.

Similarly to [3], the proposed method includes the classification of the occurring words of the fixed length into classes, however, the classification procedure as well as further methodology is different in our work. Instead of classifying the words based on the amount of randomness in them, our merging criterion for n -grams is based on similarities in probabilities of n future transitions. Therefore, instead of inferring syntactical classes, our approach infers not only classes of statistically similar words but also connections between them, i.e., dynamics in the form of a generative automaton, allowing further syntactical analysis of the grouped words and probabilistic transitions between them. While for epsilon-machine construction, n -gram length is pre-defined, in Microsynt the optimal word size is inferred from data. The automatic inference of model's parameters could potentially become further modification, although the optimality condition for the proposed method is non-obvious and non-trivial to define.

The epsilon-machine method assumes the time stationarity of the given sequences, therefore for the sequences of shorter duration, it is recommended to explicitly test their stationarity prior to model construction. In our experimental study, stationarity was assumed for the 3-minute recordings within a cognitive mode at a group level, which was not tested, but is likely to hold due to the tendency towards increase in stationarity at longer time scales [69].

The presented epsilon-machine reconstruction method is based on the words of length n occurring in data and empirical transition probabilities, conditioned on the last n microstates. It is thus guaranteed to capture temporal dependencies of the sequences that are of size n or shorter (by merging the corresponding n -grams with different prefixes into one causal state such that prefixes become irrelevant). This already aligns with the finding of [69] that the EEG microstate sequences description generally requires longer memory than just the two most recent microstates, but only finite memory in the long run. In the experimental study the word size up to 7 was used and was found to be sufficient for distinguishing sequences from different sources, however, the question of how well the model with $n < 8$ captures the temporal dependencies of the data was not studied. This can be the direction of further research, assessed by methods similar to [69] and [28], and can potentially help in establishing the optimal history length size. However, whether the presented model or similar DPFA reconstruction techniques are capable of capturing temporal dependencies of the sequences of size longer than n remains a question. Intuitively it is possible with the epsilon-machine inference methods that do not directly rely on finite words occurring in data, e.g., topological reconstruction [61], but not possible for

the methods that do. However, there is evidence [58], also recreated in Chapter 2, that an epsilon-machine built based on n -grams by CSSR method is capable of reconstructing (the exact structure of) even a non-Markovian process with infinite memory, such as the so-called even process with unbounded dependencies. This is possible since epsilon-machines do not encode memory of the past in the last n -observations, but rather in causal states. Theoretically, given an infinite amount of data and using infinite histories, the epsilon-machine method can accurately infer a correct model of the underlying process generating the sequence of observations [16]. However, in practical settings, data set size is always limited, as well as the history length, and therefore every reconstructed epsilon-machine is only the approximation of the target minimal generative dynamical system. Moreover, using histories of length n , the state space grows exponentially in the n -gram length, so even for the large datasets the maximal computationally feasible n -gram length may constrain the accuracy of the reconstructed model. The above comprises a limitation of the suggested approach, however, the success in differentiating conditions and groups obtained in our study demonstrates that essential differences in dynamical features can be captured with the proposed method (i.e., are sufficiently approximated), even when using relatively small dataset sizes and short word lengths $n < 8$.

In conclusion, the conducted study demonstrates the potential of applying the dynamical model ϵ -machine (based on both complex EEG microstate syntax and n -gram transition probabilities) for distinguishing EEG microstate sequences of different groups and cognitive conditions. Accomplishing the primary objective, the presented analysis found evidence for a strong separability between groups, while making partial progress toward the secondary objective of separating cognitive modes, as evidenced by considerably high separability of these modes within non-meditators.

Chapter 4

The Study of the Transformation Semigroup of the Abelian and Directed Non-Abelian Sandpiles

The sandpile model is a discrete event dynamical system always considered on a fixed graph with certain toppling rules for the stabilization of the configurations of the graph. In this chapter we first consider the Abelian sandpile on a 2-regular graph: show the structure of the recurrent configurations, idempotent, and generators of the recurrent group, and conclude it is cyclic; additionally, we consider the complexity of the arbitrary finite Abelian semigroup. Second, we consider the non-Abelian aperiodic sandpile semigroup on directed rooted trees: show the aperiodic complexity of the semigroup and construct the emulation of the sandpile semigroup by a minimum length wreath product of irreducible blocks.¹

4.1 Introduction

The sandpile is a dynamical system defined on a fixed graph, where every vertex is a reservoir that has the capacity to contain up to a certain amount of sand grains, and every edge is a path along which the grains can move between the vertices. The dynamics arise from the process of adding the grains to randomly selected vertices until some vertex of the graph becomes overfilled, i.e., exceeds its capacity, and then topples its sand grains to its

¹The results described in this chapter, except Section 4.3.3, were presented at the VI AMMCS International Conference on Addressing Modern Challenges in the Mathematical, Statistical, and Computational Sciences and accepted for publication in the conference proceedings book [20].

neighbors. After enough time, when many vertices of the graph are close to being full, one added grain can cause a long sequence of firings before stabilizing the graph. Such wave of topplings is referred to by physicists as avalanches, and this property of the nonlinear dynamical system - to gradually approach a critical point where a minor change causes a drastic transition - is called self-organized criticality. Examples of such critical-point phenomena from nature can be the phase transition, earthquakes, or the fluctuations in economic systems such as financial markets, while the sandpile is the simple abstract model that demonstrates it, and is a good example of how the natural complexity emerges from the simple local interactions. The sandpile model was first introduced on two-dimensional infinite square grids by Bak, Tang, and Wiesenfeld [5], expressing this physical phenomenon with a cellular automaton. The early theoretical work showed that such a process obeys power-law distributions (in terms of how many vertices topple in avalanches) and can be linked to fractal geometry and pink (1/f) noise. The other physical properties of the system, such as the conservation of energy, were also investigated.

If such a model is considered on the finite graph with a sink vertex, every unstable (overfilled) configuration can be stabilized in a finite number of steps. It was shown that the order in which the vertices are toppled does not matter and the resulting stable configuration is unique. It appeared in the literature under the names of dollar game [7, 8] and chip-firing [34], and was studied from algorithmic perspective [63, 44, 66], for example, on finite graphs without a sink considering whether an avalanche terminates and what is a bound on its length.

Another view on the sandpile model considers only the resulting configuration as a subject of interest, disregarding all the intermediate steps of the stabilization process, i.e., it studies the so-called sandpile semigroup. The stable configurations of sand grains in the graph can be considered as the state space acted upon by the operations of adding a grain at any vertex and stabilizing the graph, giving the transformation semigroup. Alternatively, the stable configurations with the associative binary operation of addition, are themselves considered a semigroup. It was shown that such a model is Abelian [21] and that every sandpile semigroup has a subgroup formed by recurrent graph configurations. The Abelian sandpile group was the subject of many different studies, particularly the connection between the graph and the group size was established through the graph Laplacian and the structure of the recurrent group has been completely determined for some families of graphs, for instance, [18, 29]. There are also more general studies about the sandpile semigroup, for example, [65], considering the directed graphs and connection between the lattice of idempotents and strongly connected components of the graph. There are also several other variations of the model, such as stochastic sandpiles [32, 12] or non-Abelian sandpiles on directed graphs [4], that, for example, consider the rate of convergence to

stationarity.

In this chapter, some properties of the transformation semigroup of the sandpile model will be considered. In section 4.3, we consider Abelian sandpiles for the simplest graph: a cycle. We first prove several stabilization properties and then describe the structure of the identity element and a generating configuration of the recurrent group. This yields an explicit alternative way of understanding the well-known cyclic structure of the recurrent group of this 2-regular graph. In section 4.4 we consider non-Abelian sandpile on rooted trees defined in [4]. We construct a minimal embedding of a sandpile monoid in a wreath product of Cartesian products of flip-flop semigroups and conclude that the aperiodic complexity [47] of the sandpile semigroup is the depth of the tree.

4.2 Preliminaries

The background on the transformation semigroups and wreath products can be found in [22], while the main definitions are given below.

Definition 4.2.1. *Let S and T be the transformation semigroups acting on the sets of states X and Y respectively, denoted as (X, S) and (Y, T) . The **wreath product** of two transformation semigroups (Y, T) and (X, S) denoted as $(Y, T) \wr (X, S)$ is a transformation semigroup $(Y \times X, W)$ that acts on the direct product $Y \times X$ of the state sets by the transformations $w \in W$ composed of two components satisfying the following:*

$$\forall w \in W, w = (f(x), s), \quad \text{such that } s \in S, \text{ and } f : X \rightarrow T,$$

where for every element $w \in W$ both the function $f : X \rightarrow T$ and the transformation $s \in S$ are uniquely determined and depend only on w . The semigroup W consists of all such possible transformations. Then for every state, the wreath product transformation acts component-wise:

$$\forall (y, x) \in Y \times X, (y, x) \cdot w = (y \cdot f(x), x \cdot s) \in Y \times X.$$

The above definition naturally extends to the wreath product of n transformation semigroups (X_i, S_i) , where every component f_i of the transformation $w \in W$ is a function of the suffix (x_{i-1}, \dots, x_1) of the state $(x_n, \dots, x_1) \in X_n \times \dots \times X_1$ that w is acting on:

$$(X_n, S_n) \wr \dots \wr (X_1, S_1) = (X_n \times \dots \times X_1, W), \quad \forall w \in W, w = (f_n, \dots, f_2, f_1),$$

where the components f_i act component-wise on $x_i \in X_i$ and are defined as follows:

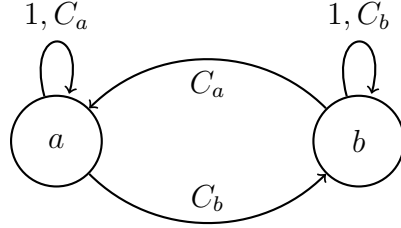
$$f_1 \in S_1, f_2 : X_1 \rightarrow S_2, f_i : X_{i-1} \times \dots \times X_1 \rightarrow S_i, \quad 1 < i \leq n.$$

Definition 4.2.2. The transformation semigroup (X, S) **divides** the transformation semigroup (X', S') , denoted as $(X, S) \preceq (X', S')$, if there exist $Y \subseteq X'$, $T \subseteq S'$, a surjective mapping of states $\alpha_2 : Y \rightarrow X$ and a homomorphism of semigroups $\alpha_1 : T \rightarrow S$ such that

$$\forall y \in Y, t \in T, \quad \alpha_2(y \cdot t) = \alpha_2(y) \cdot \alpha_1(t).$$

For the above, it is equivalent to say that (X', S') **emulates** (X, S) . If the mapping of states α_2 is bijective and α_1 is a semigroup isomorphism, it is said that (X, S) embeds in (X', S') , denoted as $(X, S) \hookrightarrow (X', S')$.

Definition 4.2.3. The **flip-flop transformation semigroup** denoted as $FF = (X, S)$ consists of three transformations $S = \{1, C_a, C_b\}$ - identity map and two constant maps, acting on two states $X = \{a, b\}$ as follows:



(a) Semigroup structure

\cdot	1	C_a	C_b
1	1	C_a	C_b
C_a	C_a	C_a	C_b
C_b	C_b	C_a	C_b

(b) Multiplication table

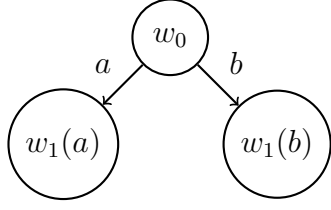
Figure 4.1: The structure and multiplication table of flip-flop transformation semigroup.

To illustrate the notion of the wreath product given above consider the following example. The wreath product of two flip-flops $FF \wr FF = (Y, W)$ acts on the states $y = (y_1, y_0) \in Y = \{a, b\} \times \{a, b\}$ by the cascade transformations $w \in W$, $w = (w_1, w_0)$ where $w_0 \in \{1, C_a, C_b\}$ and $w_1 : \{a, b\} \rightarrow \{1, C_a, C_b\}$ as $y \cdot w = (y_1 \cdot w_1(y_0), y_0 \cdot w_0)$.

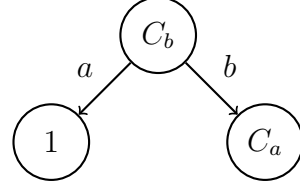
Every transformation of this wreath product $FF \wr FF$ can be represented as a cascade as shown in Figure 4.2a, where nodes describe the transformations of the corresponding level and edges describe the dependencies on states. Consider the lexicographic ordering of states in Y , i.e., $y_1 = (a, a)$, $y_2 = (a, b)$, etc. Then the transformation $t = (w_1, w_0) \in W$, illustrated in Figure 4.2b, where $w_0 = C_b$, $w_1(a) = 1$, $w_1(b) = C_a$ acts on the first state $y_1 = (a, a)$ as follows:

$$y_1 \cdot t = (a, a) \cdot (w_1, w_0) = (a \cdot w_1(a), a \cdot w_0) = (a \cdot 1, a \cdot C_b) = (a, b) = y_2, \quad \text{i.e. } y_1 \mapsto_t y_2,$$

i.e., mapping the first state y_1 to the second state y_2 . In list transformation notation t can be described as a list where $t_i = j$ means $y_i \mapsto_t y_j$ for fixed ordering of states. Thus, for the illustrated transformation t the list notation gives $t = [2, 2, 4, 2]$.



(a) Transformation structure



(b) Example: $t = [2, 2, 4, 2]$

Figure 4.2: The general structure of cascade transformations $w = (w_1, w_0) \in W$ of the wreath product $FF \wr FF = (Y, W)$ of two flip-flop semigroups $FF = (X, S)$, where $Y = X \times X$. The example of such transformation $t \in W$ is given in cascade representation and list transformation notation, where $t_i = t(y_i)$ for lexicographically ordered states $y_i \in Y$.

The concepts of wreath product and emulation/division defined for transformation semigroups (X, S) can also be considered in general for any semigroup S , assuming it acts on itself (S, S) . Every finite semigroup can be emulated by a wreath product of some building blocks (smaller semigroups and groups), called a wreath product decomposition of S . For instance, the flip-flop semigroup defined above is the simplest non-trivial aperiodic building block that can be used in such decomposition. As decomposition allows us to understand what simple components can represent the system's structure, a complexity measure that maps finite transformation semigroups to natural numbers arises from it.

Definition 4.2.4. [64] Consider a wreath product decomposition of finite semigroup S :

$$S \preceq A_n \wr G_n \wr \cdots \wr G_1 \wr A_0,$$

where G_i are groups and A_i are aperiodic semigroups (i.e., do not contain any non-trivial subgroups). The **Krohn-Rhodes complexity** of semigroup S , denoted as $cpx(S)$ is the smallest number n of non-trivial groups required to emulate the semigroup S .

The complexity measure cpx is the unique point-wise maximal function that satisfies the following axioms:

1. $cpx(G) = 1$ for any group G ,
2. $cpx(FF) = 0$ for flip-flop FF ,
3. $cpx(S \wr T) \leq cpx(S) + cpx(T)$,
4. $cpx(S \times T) \leq \max(cpx(S), cpx(T))$,
5. $S \preceq T \implies cpx(S) \leq cpx(T)$.

Every aperiodic semigroup can be emulated by the wreath product of flip-flops, which gives complexity zero by the above definition. To study finite aperiodic semigroups, con-

sider the modified definition of complexity function: define the complexity of flip-flop semigroup as one, denote the direct product of k flip-flops as U_k , then the definition is:

Definition 4.2.5. [47] For a finite aperiodic semigroup S , the **aperiodic complexity** function, denoted as $C(S)$, is defined as the minimum number of levels n in the wreath product necessary to emulate the system:

$$S \preceq A_n \wr \cdots \wr A_1, \quad \text{where } A_i = FF \text{ or } A_i = U_k = \underbrace{FF \times \cdots \times FF}_k, \text{ for some } k.$$

The function C is the unique maximal function on aperiodic semigroups that satisfies $C(FF) = 1$ and axioms 3,4,5 for Krohn-Rhodes complexity.

The additional facts from the semigroup theory that will be used later are given below.²

Definition 4.2.6. The semigroup morphism $\phi : S \rightarrow T$ is called **aperiodic** if for all idempotents $e \in T$ the inverse image $\phi^{-1}(e)$ is an aperiodic semigroup.

Statement 4.2.7. For any homomorphism $\phi : S \rightarrow T$ between semigroups S and T the following properties are equivalent (proposition in [25], p.334): (i) ϕ is aperiodic, (ii) if G is a group in S and $G\phi$ is a singleton, then G is a singleton, (iii) ϕ is injective on groups in S .

Statement 4.2.8. Let S be a finite semigroup. For an idempotent $e \in S$, $e^2 = e$ denote maximum size subgroup of S containing the idempotent e as G_e . The following holds [53]³:

- a) every element $s \in S$ has a unique idempotent power s^ω , i.e., $s^\omega s^\omega = s^\omega$, and there is the integer $\omega \geq 1$ such that $s^\omega s^\omega = s^\omega$ is the idempotent for all elements $s \in S$,
- b) the identity element of the cyclic group $C = \langle s \rangle$ generated by any element $s \in S$ is the idempotent power s^ω , and $s^{\omega+k} \in C$ for every integer $k \geq 0$,
- c) every group $G \subseteq S$ whose identity is $e \in G$, $e^2 = e$ is contained in a maximal group $G \subseteq G_e$ of this idempotent and for two idempotents $e_1 \neq e_2$ the corresponding maximal groups are disjoint $G_{e_1} \cap G_{e_2} = \emptyset$.

Lemma 4.2.9 (Fundamental Lemma of Complexity [64]). If $\phi : S \rightarrow T$ is an aperiodic homomorphism of semigroups from S to T , then $\text{cpx}(S) \leq \text{cpx}(T)$.

²These are known facts not the results of this thesis (statements 4.2.7, 4.2.8, lemma 4.2.9). The rest of the stated lemmas and theorems are all derived/reproved within the scope of this study.

³For a list of general facts the book with corresponding background on semigroups is given as a reference.

4.3 Abelian sandpiles

A detailed description of the Abelian sandpile model and its properties can be found in [21]. Here we list several key features, needed for understanding the work in this thesis.

4.3.1 Model description

In an arbitrary finite non-directed graph $\mathcal{G} = (V, E)$ fix one vertex, reachable from all other vertices⁴, to be the sink—a dissipative vertex. The capacity c_i of every non-sink vertex equals its degree (the number of edges it is connected to). The sink vertex is always empty and absorbs all grains that fall into it. For a fixed order of vertices, a configuration of the sandpile graph can be described by the vector (a_0, a_1, \dots, a_N) where $|V| = N + 1$, and a_i denotes the number of grains in the vertex v_i . A configuration is called unstable if the number of grains in at least one of the vertices contains at least as many grains as its capacity, i.e., $\exists 1 \leq i \leq N$ such that $a_i \geq c_i$, otherwise it is called stable. Stabilization is a finite iterative process of toppling all unstable vertices simultaneously, which eventually leads to a unique stable configuration. Here the toppling of an unstable vertex refers to the operation of passing one grain to each of its neighbors.

Definition 4.3.1. *The stable sandpile configuration is called **recurrent** if it is reachable from every other stable configuration by adding several sand grains to some vertices and stabilizing the graph. Otherwise, it is called **transient**.*

Thus all stable configurations are divided into these two disjoint types. The example of a recurrent configuration is one where every vertex contains the maximum amount of sand grains, i.e., $a_i = c_i - 1$. Actually, for every stable configuration (b_0, \dots, b_N) , $b_i \leq c_i - 1$ there exists a complement configuration (f_0, \dots, f_N) such that $f_i = c_i - b_i$ for $1 \leq i \leq N$ that is also stable as $0 \leq f_i \leq c_i - 1$, and it gives maximal configuration, when added to (b_0, \dots, b_N) . Thus, showing that the maximal configuration $a_i = c_i - 1$ is recurrent according to the above definition.

Definition 4.3.2. *The set of all stable sandpile configurations with the binary operation of adding configurations vertex by vertex and stabilizing the result, is associative and closed under composition, therefore forming the **sandpile semigroup**. The subset of all recurrent configurations is a minimal ideal of the sandpile semigroup that forms the sandpile group, sometimes called the **recurrent group**.*

⁴Thus here and later graphs are assumed to be connected.

Definition 4.3.3. A non-empty subset I of a commutative semigroup S is called an **ideal** if $SI \subseteq I$, i.e., $i \in I, s \in S$ implies $si \in I$. An ideal M of S is minimal if every ideal I of S included in M coincides with M , i.e., $I \subseteq M \implies I = M$. The minimal ideal of a commutative semigroup, when it exists, is a group.

As shown above, every sandpile has at least one recurrent configuration, thus it also has at least the trivial sandpile group.

4.3.2 One-dimensional case: 2-regular non-directed graph

Consider a two-regular graph $\mathcal{G} = (V, E)$ which has $|V| = N + 1$ vertices and v_0 is the only sink vertex. Such graph has *circle structure*, and in a stable configuration, each vertex can contain at most one sand grain. The sandpile semigroup defined by such a graph is mentioned in [52] as a one-dimensional model. They give an intuitive formulation of the stabilization rule by mirroring unstable vertex. This rule is given without proof, so first we formalize and prove this statement and then use it to describe the sandpile group of a circle graph.

Lemma 4.3.4. Consider the Abelian sandpile model on a 2-regular graph $\mathcal{G} = (V, E)$, $|V| = N + 1$ with a sink vertex v_0 . Let the vector $a = (a_0, a_1, \dots, a_N, a_{N+1})$ describe a configuration of grains with $a_0 = a_{N+1} = 0$ corresponding to the sink vertex. Consider adding a grain into the vertex v_s such that $0 < s < N + 1$, $a_s = 1$. Let $i < s$ and $j > s$ be the indices of the two closest empty vertices, that is $0 \leq i < s < j \leq N + 1$, $a_i = a_j = 0$, $a_k = 1$ for all $i < k < j$. Then, in the result of stabilization of the configuration a with $a_s = 2$ obtain the configuration $\bar{a} = (\bar{a}_0, \bar{a}_1, \dots, \bar{a}_N, \bar{a}_{N+1})$ constructed as follows:

1. set $\bar{a}_s = 1$ and $\bar{a}_i = \bar{a}_j = 1$, then $\bar{a}_{i+j-s} = 0$,
2. set $\bar{a}_k = a_k$ for the rest of the vertices, i.e., $\forall k$ such that $k \notin \{i, j, s, i + j - s\}$,
3. set $\bar{a}_0 = \bar{a}_{N+1} = 0$ for the sink vertex.

Proof. Consider induction for $(j - i)$. Ensure that in two base cases statement holds:

$$(j - i = 2) : \dots 020 \dots \rightarrow \dots 101 \dots, \quad (j - i = 3) : \dots 0210 \dots \rightarrow \dots 1020 \dots \rightarrow \dots 1101 \dots,$$

where \bar{a} is stabilized in one or two updates. Assume it is true for all $j - i \leq M$ and consider $j - i = M + 1$. W.l.o.g. $i < s - 1$, after the first toppling $a_s = 0$, $a_{s-1} = 2$.

- if $s + 1 < j$, then $\overline{a_{s+1}} = 2$. Using induction hypothesis, first stabilize segment $\overline{a_i}, \dots, \overline{a_s}$ - get $\overline{a_{i+1}} = 0$, then stabilize segment $\overline{a_{i+1}}, \dots, \overline{a_j}$ - get $\overline{a_{i+j-s}} = 0$:

$$\bar{a} = (\dots \underset{i}{0}1\dots \underset{s}{1}2021\dots \underset{j}{1}0\dots) \longrightarrow (\dots \underset{i}{1}01\dots \underset{s}{1}1121\dots \underset{j}{1}0\dots) \longrightarrow (\dots \underset{i}{1}1\dots \underset{i+j-s}{1}0 \quad \underset{j}{1}\dots \underset{j}{1}1\dots),$$

- otherwise $s + 1 = j$, $\overline{a_j} = 1$ and stabilization of $\overline{a_i} \dots \overline{a_s}$ results in a stable configuration, with $\overline{a_{i+1}} = 0$, where $i + 1 = i + j - (j - 1) = i + j - s$ as expected.

The third step in the lemma ensures that grains are discarded in the new configuration, if fall into the sink, i.e., if $i = 0$ and/or $j = N + 1$.

□

The structure of the sandpile group

For convenience, let's denote the configuration containing the maximum possible number of grains (maximal configuration) by A , and for every $1 \leq i \leq N$ the configurations where exactly only the i -th vertex is filled by C_i and where exactly only the i -th vertex is empty by B_i . Describing them by binary vectors $a = (a_0, a_1, \dots, a_N, a_{N+1})$ with $a_0 = a_{N+1} = 0$ corresponding to the sink vertex, we get the following:

$$A = \begin{pmatrix} 01\dots 1 & 0 \\ 0 & \underset{N+1}{} \end{pmatrix}, \quad B_i = \begin{pmatrix} 01\dots 101\dots 1 & 0 \\ 0 & \underset{i}{} \quad \underset{N+1}{} \end{pmatrix}, \quad C_i = \begin{pmatrix} 00\dots 010\dots 0 & 0 \\ 0 & \underset{i}{} \quad \underset{N+1}{} \end{pmatrix}, \quad i \in \{1, \dots, N\}.$$

Because of the circle structure of the 2-regular graph extend this notion for the negative indices: $B_{i-k} = B_{(i-k+N+1)}$. Also, notice the equality $B_0 = B_{N+1} = A$ as both a_0 and a_{N+1} correspond to the sink. Using the stabilization rule described in Lemma 4.3.4, and the defined above notation for configurations, the following addition rules can be derived for the sandpile on the circle graph.

Lemma 4.3.5. *For any indices $1 \leq m, k \leq N$ the following holds:*

$$B_m + C_k = B_{m-k} = B_{N+1+m-k}.$$

Proof. To stabilize addition using Lemma 4.3.4 have to establish the closest empty vertices i and j . Consider cases:

1. $k < m$, then $i = 0$, $j = m$, and $B_m + C_k = B_{m-k}$,

2. $k = m$, then $B_m + C_k = B_m + C_m = A = B_0 = B_{N+1}$ or equivalently $A = B_{m-k} = B_{N+1+m-k}$,
3. $k > m$, then $i = m$, $j = N + 1$, and $B_m + C_k = B_{N+1+m-k}$.

□

Lemma 4.3.6. *The addition of two grains of sand to the vertices equidistant from the sink acts as identity transformation on the type A and type B configurations:*

$$\forall k < N - k + 1 : \quad A + C_k + C_{N-k+1} = A, \quad B_m + C_k + C_{N-k+1} = B_m \quad 1 \leq m \leq N.$$

Proof. By Lemma 4.3.4 have: $(A + C_k) + C_{N-k+1} = B_{N-k+1} + C_{N-k+1} = A$. By Lemma 4.3.5 have: $B_m + C_k + C_{N-k+1} = B_{m-k} + C_{N-k+1} = B_{N+1+m-k-(N+1-k)} = B_m$. □

The above rules for configuration addition help to understand the structure of the sandpile group and, as stated in the following Theorem 4.3.7, show which configurations are generators and which act as identity.

Theorem 4.3.7. *For the Abelian sandpile on a 2-regular graph $\mathcal{G} = (V, E)$ with $|V| = N+1$ vertices where the only sink is the vertex $v_0 \in V$ the following hold:*

1. *The elements of the sandpile group $G = G(\mathcal{G})$ are the configurations that have at most one empty vertex:*

$$G = \{A, B_1, \dots, B_N\}.$$

2. *The identity element $I = I(\mathcal{G})$ of the sandpile group $G = G(\mathcal{G})$ has the following structure:*

$$I = \begin{cases} A, & \text{if } N \text{ is even,} \\ B_q, & q = \frac{N+1}{2}, \text{ otherwise.} \end{cases}$$

3. *A generator $\tilde{G} = \tilde{G}(\mathcal{G})$ of the group can be defined as follows:*

$$\tilde{G} = \begin{cases} B_2 & - \text{if } N = 4k + 1 \text{ for some } k, \\ B_1 & - \text{otherwise.} \end{cases}$$

4. *The sandpile group $G = G(\mathcal{G})$ is a cyclic group of size $N + 1$.*

Proof. 1. Graph \mathcal{G} has $N+1$ spanning trees, so there are $N+1$ recurrent configurations [21]. The maximal configuration A is reachable from any other configuration, so is recurrent by definition. Moreover, by Lemma 4.3.4 we have $B_i = A + C_{N+1-i}$, so B_i is recurrent as well. So, A and B_i are all elements of the sandpile group $G = G(\mathcal{G})$.

2. For $N = 2m$ identity is $I = A = \sum_{k=1}^m (C_k + C_{N-k+1})$. Also, for $N = 2m+1$ we have $I = B_{m+1} = \sum_{k=1}^m (C_k + C_{N-k+1})$, so Lemma 4.3.6 implies that $A + I = A$, $B_i + I = B_i \implies I$ is the identity element.

3. First, consider $N = 2m$ and show that $\tilde{G} = B_1$:

As $B_1 = C_2 + \dots + C_N = \sum_{k=2}^m (C_k + C_{N-k+1}) + C_N \implies B_i + B_1 = B_i + C_N$ using Lemma 4.3.6. Then conclude $B_i + B_1 = B_{i+1}$, $1 \leq i \leq N$ using Lemma 4.3.4. So, B_1 generates all elements of the group $\langle B_1 \rangle = \{B_1, B_2, \dots, B_N, B_{N+1} = A\} = G$.

Second, consider $N = 4m+1$ and show that $\tilde{G} = B_2$:

$$B_2 = \sum_{k=1, k \neq 2}^{2m} (C_k + C_{N-k+1}) + C_{N-1} + C_{2m+1} \implies B_i + B_2 = B_i + C_{4m} + C_{2m+1},$$

for any $i \in \{1, \dots, N\}$, using Lemma 4.3.6. Then we conclude that $B_i + B_2 = B_{i-4m+N+1} + C_{2m+1} = B_{i+2} + C_{2m+1} = B_{i-(2m-1)}$ using Lemma 4.3.5. In other words, by adding a configuration B_2 to configuration B_i we move the position of the empty vertex i by $2m-1$ positions backward in a circle. Repeating additions of B_2 and considering the intermediate moment $B_i + k_1 \cdot B_2$, then $\exists k_2 : B_i + k_1 \cdot B_2 = B_i + (k_1 + k_2) \cdot B_2$ as we have a finite number of type B configurations. Suppose that the configuration repeats after t full turn around the circle, then

$$(2m-1)k_1 = (2m-1)(k_1 + k_2) - t(N+1) \implies k_2(2m-1) = 2t(2m+1).$$

As $2m-1$ and $2m+1$ are coprime $\implies k_2 | (4m+2) \implies k_2 \geq N+1 \implies$ at least $N+1$ different elements occur, before the sequence $B_i + B_2 + \dots + B_2$ has repeated element. So, $\langle B_2 \rangle = G(\mathcal{G})$.

The case where $N = 4m+3$ is analogous, so omitted.

4. As G has $N+1$ elements and one generator, it is cyclic group $G = C_{N+1}$.

□

To illustrate the pattern that the identity and generators follow, consider Figure 4.3 where the structure of sand configurations described by Theorem 4.3.7 is represented by

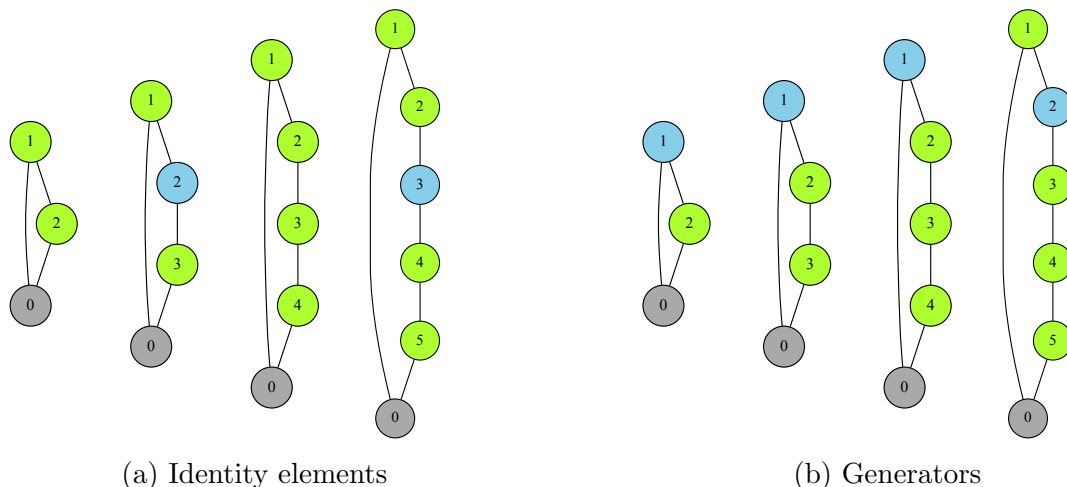


Figure 4.3: The illustration of the structure of the identity elements and generators for a circle graph with $N \in \{2, 3, 4, 5\}$ non-sink vertices. The color indicates the number of sand grains in the corresponding vertex, and the number in the vertex denotes the ordering. The blue color denotes an empty vertex, the green denotes one containing a single sand grain, and the gray vertex is a sink.

colors on the circle graph (narrowed graphs for compactness) with fixed vertex ordering. With the increase in the number of vertices, the identity element follows the alternating pattern, while the generator has a period of four steps. It is worth noting, that while the cyclic group C_n is generated by one element, the generator configuration is not unique. The one shown in the example and described in the theorem is the lexicographically smallest.

4.3.3 Complexity of Abelian semigroups

The following fact about the complexity of Abelian semigroups was briefly stated, for example in [53] and [38] without giving the formal proof. Although it is a known fact and was probably considered by the authors as the natural implication of the described theory, this question arose during the study of Abelian sandpiles, and two versions of proof were derived.

Theorem 4.3.8. *The Krohn-Rhodes complexity $cpx(S)$ of any Abelian semigroup S is at most one, $cpx(S) \leq 1$.*

Proof. Let $E \subseteq S, |E| = m$ be the set of all idempotents of S with a fixed numbering of elements $E = \{e_1, e_2, \dots, e_m\}$ and G_k denote the maximal subgroup of S containing the

idempotent e_k (some of the G_k may be trivial groups). Due to Statement 4.2.8(c) all G_k are disjoint. Denote the union $T = G_1 \sqcup \cdots \sqcup G_m$.

Due to Statement 4.2.8(a) there is an integer ω such that $\forall s \in S$, s^ω is the idempotent. Consider a semigroup morphism $\phi : S \rightarrow Im(\phi) \subseteq S$ defined as $\phi(s) = s^\omega s$, $\forall s \in S$. As multiplication in S commutes, ϕ is a structure preserving mapping: $\forall s_1, s_2 \in S$, $\phi(s_1)\phi(s_2) = s_1^\omega s_1 s_2^\omega s_2 = s_1^\omega s_2^\omega s_1 s_2 = (s_1 s_2)^\omega s_1 s_2 = \phi(s_1 s_2)$. For any element $s \in S$, suppose $s^\omega = e_k$, then its image $\phi(s) = s^\omega s = s^{\omega+1} \in \langle s \rangle \subseteq G_k$ belongs to the corresponding group G_k , due to Statement 4.2.8(b),(c), meaning $Im(\phi) \subseteq T$. Moreover, every group element $s \in G_k$ is mapped to itself $\phi(s) = s^\omega s = e_k s = s \in G_k$, meaning that $T \subseteq Im(\phi)$ and ϕ is injective on groups. The mutual inclusion implies $Im(\phi) = T$, and T is a subsemigroup of S . Thus we conclude that ϕ is a semigroup homomorphism $\phi : S \rightarrow T$ that is injective on groups in S , therefore aperiodic due to Statement 4.2.7. The Fundamental Lemma of Complexity 4.2.9 implies $cpx(S) \leq cpx(T)$.

Consider a new element I acting as identity on S , that is $\forall s \in S \cup \{I\}$, $sI = Is = s$. Let $Y = G_1^I \times \cdots \times G_m^I$ with component-wise multiplication denoted by $*$, where $G_i^I = G_i \cup \{I\}$. For the generator g of group $G_k \subseteq T$ there is the element $\vec{y}_g = (y_1, \dots, y_m) \in Y$ such that $y_i = g$ for $i = k$ and $y_i = I$ for $i \neq k$. Let Y' be the semigroup $Y' \subseteq Y$ generated by elements \vec{y}_g for all generators $g \in G_k$ and all groups $1 \leq k \leq m$. Consider the mapping $\psi : Y' \rightarrow T$ defined for all $\vec{y} = (y_1, \dots, y_m) \in Y'$ as $\psi(\vec{y}) = \prod_{j=1}^m y_j$ with multiplication in S . As S is Abelian, for all $\vec{a} = (a_1, \dots, a_m), \vec{b} = (b_1, \dots, b_m) \in Y'$ we have $\psi(\vec{a})\psi(\vec{b}) = \prod_{j=1}^m a_j \prod_{j=1}^m b_j = \prod_{j=1}^m (a_j b_j) = \psi(\vec{a} * \vec{b})$. Therefore, $\psi : Y' \rightarrow T$ is a surjective homomorphism that implies division $T \preceq Y$.

By the Definition 4.2.4 of Krohn-Rhodes complexity, $cpx(T) \leq cpx(Y)$ (axiom 5) and $cpx(Y) = 1$ (axiom 1). Thus, $cpx(S) \leq cpx(T) \leq cpx(Y) = 1$ completes the proof. \square

The above proof gives an understanding of the semigroup's structure by using groups G_i as building 'blocks' for mappings. If the homomorphism ψ could be constructed, that is not only surjective but injective (for a modified Y), that would correspond to the embedding of blocks. As this embedding was not achieved in this study, an alternative shorter proof that shows a 'rougher' emulation, but does not use the Fundamental Lemma of Complexity 4.2.9 was derived and is given below.

Proof for Theorem 4.3.8. Let m be the semigroup size $|S| = m$ and fix the ordering of elements. Consider the semigroup $Y = S_1^I \times \cdots \times S_m^I$ with component-wise multiplication denoted by $*$, where $S_j^I = \langle s_j \rangle \cup \{I\}$ is the monogenic semigroup generated by the element $s_i \in S$ and I is the element acting as identity on S , i.e., $\forall s \in S \cup \{I\}$, $sI = Is = s$.

Consider the mapping $\psi : Y \rightarrow S$ defined for all $\vec{y} = (y_1, \dots, y_m) \in Y$ as $\psi(\vec{y}) = \prod_{j=1}^m y_j$ with multiplication in S . As S is Abelian, for all $\vec{a} = (a_1, \dots, a_m), \vec{b} = (b_1, \dots, b_m) \in Y'$ we have $\psi(\vec{a})\psi(\vec{b}) = \prod_{j=1}^m a_j \prod_{j=1}^m b_j = \prod_{j=1}^m (a_j b_j) = \psi(\vec{a} * \vec{b})$. Also, for every element $s_i \in S$ there is the element $\vec{y} = (I, \dots, s_i, \dots, I) \in Y$ that is mapped to it $\psi(\vec{y}) = I^{i-1} s_i I^{m-i} = s_i$. Therefore, $\psi : Y \rightarrow S$ is a surjective homomorphism that implies division $S \preceq Y$.

Therefore, $cpx(S) \leq cpx(Y) \leq \max_i cpx(S_j^I)$ by axioms 5 and 4 of the Definition 4.2.4. As every monogenic semigroup has complexity at most one and adding an identity element does not increase it, have that $cpx(S_j^I) \leq 1$. Thus, we conclude $cpx(S) \leq 1$. \square

4.4 Non-Abelian sandpiles on the rooted trees

The majority of the sandpile models, even for the directed graphs where the movement of grains is restricted in direction, result in Abelian sandpile semigroups. In this part, we consider the variation of the model where not only the addition of grains and stabilization of vertices is possible, but also the operation of shifting the sand gains along the edges. This modification makes the generated sandpile semigroup non-Abelian, and the rest of this chapter concentrates on the structure of this type of semigroup.

4.4.1 Model description

The non-Abelian sandpile model is defined in [4] on a rooted tree $\mathcal{G} = (V, E)$, $|V| = N + 1$ with the root vertex $v_0 \in V$ being the only sink vertex and the edges directed towards the root - the only direction in which the sand grains can move. Denote the set of leaves as $L \subset V$ of the size $|L| = m$. The capacity of every vertex $v_i \in V$ is now **not** related to the out-degree but can be set to any non-negative integer $c_i \geq 0$ and $c_0 = 0$ for the sink, that only absorbs the grains. The operations allowed on the graph are of two types and they constitute the $m + N$ generators of the sandpile semigroup:

- f_{v_i} - add a sand grain to the vertex v_i and stabilize the graph; it is defined for all leaves $v_i \in L$, but not other vertices.
- g_{v_i} - lift a sand grain from the vertex v_i (if it has any) to its parent vertex (towards the root) and stabilize the graph; it is defined for all $v_i \in V$, $v_i \neq v_0$ non-root vertices. If v_i is empty, this operation does not affect the configuration.

Three features make this model different from the previously considered. First, the operation of adding the grain is restricted to the leaves only, making them the only reservoirs through which the grains can be added to the graph. Second, the additional operation of lifting a sand grain to the parent vertex is introduced. Third, the capacity of every non-sink vertex is an arbitrary non-negative integer. When considering all edges to be directed towards the root, the stabilization process is the same as before. The first two changes result in the semigroup being non-Abelian, as the operations of adding and lifting a grain do not commute. Actually, for an empty leaf, adding a grain and then lifting it up results in an empty leaf, but switching the order of operations results in the leaf containing a grain.

4.4.2 Embedding of the sandpile with equal to one capacity

For the rooted tree $\mathcal{G} = (V, E)$, $|V| = N + 1$ we denote the depth of the tree by $D = D(\mathcal{G})$. For each level $i \in \{1, \dots, D\}$ of the tree we denote the number of the vertices in the tree which are located on the depth i by m_i , so $M = \{m_1, m_2, \dots, m_D\}$, $\sum_{i=1}^D m_i = N$. The constructive proof of the following theorem derived in this work is described throughout this section.

Theorem 4.4.1. *The aperiodic complexity of the sandpile semigroup $S = S(\mathcal{G})$ acting on the configurations $X = X(\mathcal{G})$ of the rooted tree $\mathcal{G} = (V, E)$ with equal to one capacity $c_i = 1$, $i \in \{1, \dots, N\}$, equals the depth $D = D(\mathcal{G})$ of the tree \mathcal{G} . Also, (X, S) can be embedded into the wreath product of D blocks of flip-flops, where each i^{th} block is a direct product of m_i flip-flops:*

$$(X, S) \hookrightarrow \underbrace{(FF \times \dots \times FF)}_{m_D} \wr \dots \wr \underbrace{(FF \times \dots \times FF)}_{m_1} = (Y, W), \quad (4.1)$$

where, D is the minimal number of blocks in the wreath product needed for the embedding $(\phi_{state}, \phi_{trans}) : (X, S) \hookrightarrow (Y, W)$ to exist.

Base case: sandpile of the path graph

Consider the path graph $\mathcal{G} = (V, E)$, $|V| = N + 1$ shown in Figure 4.4 with *equal to one capacity*:

where the vertices are labeled as above, and we refer to the vertex labeled with i as v_i . The sandpile semigroup $S = S(\mathcal{G})$ acts on the state space $X = X(\mathcal{G})$ of the size $|X| = 2^N$. Each state $x \in X$ is a binary vector, where $x_i \in \{0, 1\}$ is the number of grains in the vertex

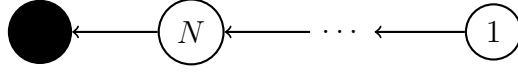


Figure 4.4: The sandpile on the path graph $\mathcal{G} = (V, E)$ with $|V| = N + 1$ labeled vertices. The root vertex drawn in black is the sink, every vertex can contain up to 1 sand grain.

v_i and let's agree on the following order of the coordinates $x = (x_N, x_{N-1}, \dots, x_2, x_1)$. The semigroup has $N + 1$ generators $S = \langle S_g \rangle$, $S_g = \{f, g_1, \dots, g_N\}$, where we denote $f = f_{v_N}$, $g_i = g_{v_i}$ for convenience.

The wreath product of N flip-flops $(FF \wr \dots \wr FF) = (Y, W)$ acts on the state space Y with the transformations $w \in W$, where each component $w_i = w_i(y_{i-1}, \dots, y_1)$ is a function of the suffix y_{i-1}, \dots, y_1 of the state y that w is acting on:

$$\forall y = (y_N, \dots, y_1) \in Y \quad y_i \in \{a, b\}, \quad |Y| = 2^N, \quad \forall w = (w_N, \dots, w_1) \in W, \quad w_i \in \{1, C_a, C_b\}.$$

Both configurations of the graph $x \in X$ and states of the wreath product $y \in Y$ have binary coordinates, so there is a natural bijection $\phi_{state} : X \rightarrow Y$ via binary strings, where $\forall x \in X$, $\phi_{state}(x)_i = a$ if $x_i = 0$ and $\phi_{state}(x)_i = 1$ otherwise. Note that i^{th} component of both $x \in X$ and $\phi_{state}(x) \in Y$ correspond to the vertex $v_i \in V$.

Denote the mapping ϕ_{trans} for every generator $q \in S_g$:

$$\phi_{trans} : S_g \longrightarrow W, \quad \phi_{trans}(q) = w^q(y_N, \dots, y_1) = (w_N^q, \dots, w_1^q) \in W.$$

Consider the components w_i^f of $\phi_{trans}(f)$ to be defined as follows:

$$w_i^f(y_{i-1}, \dots, y_1) = \begin{cases} C_b, & \text{if } i = 1 \text{ or } \forall k \in \{1, \dots, i-1\} \ y_k = b, \\ 1, & \text{otherwise.} \end{cases} \quad (4.2)$$

That is, it acts on the state space in the following way:

$$\forall y = (\underbrace{\dots}_q, \underbrace{a, b \dots b}_k) \in Y, \quad w(y) = (\underbrace{1 \dots 1}_q, \underbrace{C_b, C_b \dots C_b}_k),$$

$$y \cdot \phi_{trans}(f) = (\underbrace{\dots}_q, \underbrace{b, b \dots b}_k),$$

where the prefix of the state $y \in Y$ of length q remains unchanged.

Similarly, consider $\phi_{trans}(g_j)$, $j \in \{1, \dots, N\}$ to have the following components:

$$w_i^{g_j}(y_{i-1}, \dots, y_1) = \begin{cases} 1, & \text{if } i \leq N - j, \\ C_a, & \text{if } i = N - j + 1, \\ C_b, & \text{if } i > N - j + 1 \text{ and } y_k = b, \ i - 1 \leq k \leq N - j + 1, \\ 1, & \text{otherwise.} \end{cases} \quad (4.3)$$

Then, any state $y = (y_N, \dots, y_1) \in Y$ with $y_i = a$ acted upon by transformation $\phi_{trans}(g_i) = w^{g_i}(y)$ remains unchanged, but for the states $y = (y_N, \dots, y_1) \in Y$ with $y_i = b$ the result of action $\phi_{trans}(g_i) = w^{g_i}(y)$ is defined as follows:

$$\forall y = (\underbrace{\dots}_q, a, \underbrace{b \dots b}_p, \underbrace{\dots}_{N-i}) \in Y, \ p + q + 1 = i,$$

$$w^{g_i}(y) = (\underbrace{1 \dots 1}_q, C_b, \underbrace{C_b, \dots, C_b, C_a}_p, \underbrace{1 \dots 1}_{N-i}), \quad y \cdot \phi_{trans}(g_i) = (\underbrace{\dots}_q, b, \underbrace{b \dots b}_p, a, \underbrace{\dots}_{N-i}).$$

From such a description of transformations and their action on the states, conclude that $\phi_{trans}(f)$ and $\phi_{trans}(g_i)$ act on the state space of the wreath product in the same way as f and g_i act on the configurations of the tree $\mathcal{G} = (V, E)$, so they generate a semigroup isomorphic to the sandpile semigroup S .

Corollary 4.4.2. *Consider the rooted tree $\mathcal{G} = (V, E)$, which consists of m path graphs, glued in a root. Each i^{th} path has n_i vertices. There is the following embedding of the sandpile semigroup:*

$$(\beta_{state}, \beta_{trans}) : (X, S) \hookrightarrow \underbrace{(FF \wr \dots \wr FF)}_{n_1} \times \dots \times \underbrace{(FF \wr \dots \wr FF)}_{n_m} = (Y, W).$$

As the model only allows the grains to move towards the root, action on one path doesn't affect others. The above corollary naturally follows from the independence of the root branches but is not a minimal embedding.

Generalization for an arbitrary tree

Consider the arbitrary rooted tree $\mathcal{G} = (V, E)$ of the depth $D = D(\mathcal{G})$ with the capacity of each vertex equal to one. Denote the depth of the vertex $v \in V$ as $d(v)$, and its parent

as $p(v)$. As \mathcal{G} is a rooted tree, the parent of any vertex is unique and we can consider a chain of parents of v on the way to the root vertex:

$$P(v) = (u_{d(v)}, \dots, u_2, u_1), \quad u_1 = v, \quad u_{d(v)} = \text{root}, \quad u_{i+1} = p(u_i), \quad 1 \leq i < d(v).$$

Suppose that all non-sink vertices are labeled with integers from 1 to N , and we refer to the vertex labeled as i with v_i . Assume for all labels holds that if $v_j = p(v_i)$, then $j > i$. This implies that indices of the vertices in the parent chain are monotonically decreasing. For every state $y \in Y$ define a vector $y(v)$ of size $d(v)$, which consists only of the components of the vector $y = (y_N, \dots, y_1) \in Y$ corresponding to the parent chain $P(v)$ of the vertex v : $y(v) = (y_{P_{d(v)}}, \dots, y_{P_1})$ where $P_j = P(v)_j$ is the j^{th} ancestor of v . Every transformation - f_{v_i} or g_{v_i} is applied to some vertex v_i and the important thing to notice is that it doesn't affect the vertices other than the chain of parents $P(v_i)$ of this vertex. Denote $\bar{w}^f(y_{i-1} \dots y_1) = w_i^f(y_{i-1} \dots y_1)$ and $\bar{w}^{g_j}(y_{i-1} \dots y_1) = w_i^{g_j}(y_{i-1} \dots y_1)$ the mapping defined by equations (4.2 - 4.3) for the case of a path graph.

Consider the following mapping $\phi_{trans} : S_g \rightarrow W$ for generators of S . For adding a grain into leaf v , the i^{th} component of the transformation $\phi_{trans}(f_v) = (w_N^{f_v}, \dots, w_1^{f_v})$ define as

$$w_i^{f_v} = w^{f_v}(y_{i-1} \dots y_1) = \begin{cases} \bar{w}_i^f(y(v)_{j-1} \dots y(v)_1), & \text{if } v_i = P(v)_j \text{ for some } j, \\ 1, & \text{otherwise.} \end{cases}$$

For lifting a grain up from the vertex v , the i^{th} component of the transformation $\phi_{trans}(g_v) = (w_N^{g_v}, \dots, w_1^{g_v})$ define as

$$w_i^{g_v} = w^{g_v}(y_{i-1} \dots y_1) = \begin{cases} \bar{w}_i^g(y(v)_{j-1} \dots y(v)_1), & \text{if } v_i = P(v)_j \text{ for some } j, \\ 1, & \text{otherwise.} \end{cases}$$

On this, the existence of embedding (4.4.1) can be considered proven.

4.4.3 Generalization for a rooted tree with arbitrary capacity

Similarly one can construct the emulation of the sandpile semigroup in the case of arbitrary capacity of the vertices. We omit the formal description of the ϕ_{trans} and ϕ_{state} mappings and only give the sketch of the construction.

For the graph $\mathcal{G} = (V, E)$ consider the operation of replacing all vertices with $c_i > 1$ with a chain of c_i vertices of capacity $c'_i = 1$, as shown in Figure 4.5. Let's call it *extended tree* \mathcal{G}^* , which now has the depth $D^* = D(\mathcal{G}^*)$ equal to the maximum sum of capacities on the parent chain among all leaves. As each colored block actually corresponds to one vertex, all permutations of the grains within the block are equivalent, and only the number of grains matters. For this reason, ϕ_{state} is no longer bijective, i.e., equation (4.1) denotes emulation, where every configuration of the graph has several lifts - corresponding states in the wreath product.

For the transformation $\phi_{trans}(f_v)$ of adding the grain to the leaf, construction is exactly the same as for an arbitrary tree with capacity one, when applied to the deepest vertex of the block corresponding to this leaf.

For the transformation $\phi_{trans}(g_v)$ of lifting the grain to the parent vertex the only change will be to skip all vertices of the extended tree that correspond to the same vertex of the original graph when lifting a grain. Then, Theorem 4.3.7 is generalized for the tree of any capacity by applying it to the extended tree.

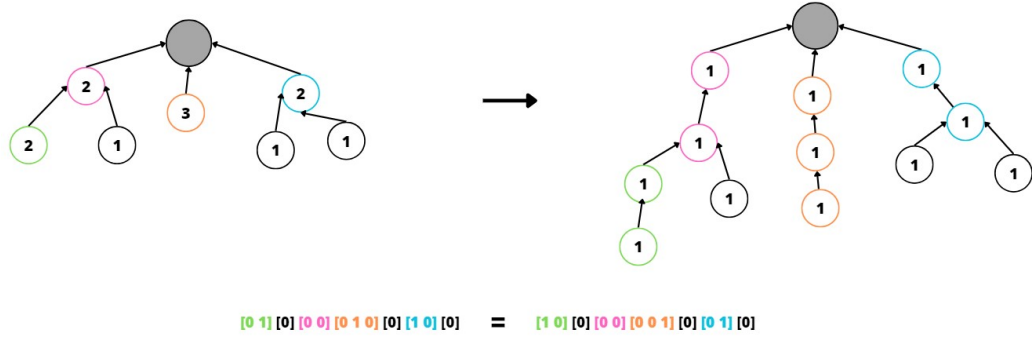


Figure 4.5: Example of the extended tree. Labels on the vertices refer to their capacities. Segments of the binary vector, corresponding to the extended vertices match the colors. The example shows the equivalence of the permutations within the blocks.

4.4.4 Aperiodic complexity of sandpile semigroups

According to [47], the aperiodic complexity $C(S)$ of the aperiodic semigroup S is the minimum number of levels in the wreath product of blocks from flip-flops necessary to emulate the system. A lower bound for aperiodic complexity is the exponent of the semigroup S_ϵ :

$$S_\epsilon = \text{least } n : s^n = s^{n+1} \quad \forall s \in S.$$

Clearly, the non-Abelian sandpile semigroup S is aperiodic. The existence of the embedding (4.1) implies the upper bound on its complexity:

$$S_\epsilon \leq C(S) \leq D.$$

Consider an arbitrary tree \mathcal{G} of capacity one and depth $D = D(\mathcal{G})$. Let v be a leaf vertex, such that $d(v) = D$, and f_v be the transformation of adding a grain to the leaf v . The exponent of f_v is D , as starting from the empty configuration we can apply it exactly D times before all vertices in the parent chain of v are full and further addition becomes a constant transformation. Then $S_\epsilon \geq D$ and

$$D \leq S_\epsilon \leq C(S) \leq D,$$

which implies their equality $C(S) = S_\epsilon = D$. So, the aperiodic complexity of the sandpile semigroup $S = S(\mathcal{G})$ is the depth of the tree $D = D(\mathcal{G})$ for the tree \mathcal{G} of capacity one, or the depth $D^* = D(\mathcal{G}^*)$ of the extended tree \mathcal{G}^* in general.

4.5 Conclusion

In this chapter, two sandpile models were considered. For the classical Abelian model, the study was restricted to the sandpile on a graph having a circle structure (the so-called one-dimensional case). First, the non-iterative ‘mirroring’ stabilization rule was proved, resulting in the rules of adding the configurations of the sandpile. Then the structure of the identity and the generator elements of the sandpile group were described in terms of the sand configurations of the graph, providing an explicit alternative way of understanding the well-known cyclic structure of the recurrent group. Also, the two ways of showing that the Krohn-Rhodes complexity of an arbitrary finite Abelian semigroup is at most one were described: with and without the fundamental lemma of complexity.

For the non-Abelian model on rooted trees, the semigroup structure was considered by relating it to the product of elementary building blocks of the semigroups. The constructive description was found to show the emulation (and embedding) of generators of the sandpile semigroup in a wreath product of flip-flops, which, moreover, is the one that contains the fewest levels in the wreath product. The existence of such an embedding (and emulation) shown by construction, allowed us to conclude on the aperiodic complexity of the sandpile transformation semigroup, which particularly equals the depth of the rooted tree.

Chapter 5

The Summary of Contributions and Further Work

Within the framework of this thesis, two automata models were considered: the deterministic-probabilistic finite automata (DPFA) and the finite automata that arise from the sandpile model. The study of the first one (DPFA), includes the formulation and implementation of the algorithm for its reconstruction from the sequences of discrete observations, the consideration of the corresponding transformation semigroup, and the application of the automaton for distinguishing EEG microstate sequences of two participant groups (meditators and controls) and three cognitive modes (mind-wandering, verbalization, and visualization). The study of the second model, sandpile automata, concentrates on the properties of the transformation semigroups only.

Chapter 2 describes an n -gram merging method targeting the reconstruction of DPFA. The contribution of this part is the design of the software that implements a variation of the automaton reconstruction method and examination of the question of the reconstructability of the permutation structures of the system from observations. The presence of the permutation subgroups in the transformation semigroup of the system is an important property, particularly for understanding the complexity of the system. To our knowledge, previous studies of automaton reconstruction have primarily been interested in learning a system that approximates the probability distribution over the generated strings well, and they do not consider the recovery of the permutation groups in practice. However, it remains an open question whether an algorithm that achieves this objective can be constructed, and the presented study takes a small step towards addressing this question. A logical next step that would complement the results presented in Chapter 2 would be to conduct a theoretical and experimental comparison of the considered algorithm and other

well-known techniques for DPFA reconstruction. The comparison could include general properties such as convergence and error bounds, as well as the group-reconstruction question raised here. A more detailed consideration could lead to understanding the novel algorithm decisions for performance improvement.

The contribution of Chapter 3 is that it develops the framework of the novel approach for using reconstructed DPFA to distinguish the EEG microstates sequences and applies the method to the data of meditators and meditation-naive healthy controls. The presented results show reliable separation of participant groups and partial achievement of the separation of cognitive modes, thus demonstrating the potential of the proposed approach. As the designed analysis framework is suitable for CSSR, it is the natural next step to compare their success in distinguishing the available EEG sequences. Similarly, other DPFA reconstruction algorithms could be utilized in such a study, with the only change being to develop and apply suitable distance metrics. Such extension of the experimental study could assist in giving other criteria to compare the reconstruction techniques and comprise a deeper development of the automata approach in the area of EEG studies. In the context of EEG microstate sequences, the important properties that the inferred model should preserve are the periodicities and long-range dependencies that are inherent in the data [69, 28]. Thus, the empirical evaluation of the reconstruction quality can be extended to assess these properties, in addition to closeness in distribution over strings and recognition of groups considered in Chapter 2.

Chapter 4 considers the transformation semigroup of the classical Abelian sandpile model on the circle graph and the modified sandpile on the arbitrary rooted trees, resulting in the non-Abelian semigroup. The presented study makes the connections between the graph and the properties of the generated sandpile semigroup. Particularly, the contribution consists of (i) for Abelian sandpile, in theoretically showing the structure of the sand grain configurations of the circle graphs which act as an identity and a generator of the recurrent sandpile group, and a new proof showing that complexity of an arbitrary finite Abelian semigroup has complexity at most one, also (ii) for a non-Abelian sandpile, in constructively showing the existence of the emulation of the sandpile semigroup by the wreath-product of direct products of flip-flops and concluding that its aperiodic complexity equals the depth of the tree. Continuing work in the same direction, a next step could be determining the structure of the identity and generators of the recurrent group for more complex graph structures. This, however, would require another approach, as due to complex and hard-to-forecast sandpile dynamics, the derivation of the non-iterative stabilization rule would be only possible in very special cases and/or simple graphs.

This concludes the summary of the results, key contributions, and suggested further direction building on the work presented in this thesis.

References

- [1] Elena Antonova, Martin Holding, Ho Chak Suen, Alex Sumich, Reinoud Maex, and Chrystopher Nehaniv. EEG microstates: Functional significance and short-term test-retest reliability. *NeuroImage: Reports*, 2(2):100089, 2022.
- [2] Elena Antonova and Chrystopher L. Nehaniv. Decoding the language of ‘now’: EEG microstates in experienced meditators, from letters to grammar. Final Scientific Report to the BIAL Foundation (Grant number: 183/16), 2023.
- [3] Fiorenzo Artoni, Julien Maillard, Juliane Britz, Denis Brunet, Christopher Lysakowski, Martin R. Tramèr, and Christoph M. Michel. Microsynt: Exploring the syntax of EEG microstates. *NeuroImage*, 277:120196, 2023.
- [4] Arvind Ayyer, Anne Schilling, Benjamin Steinberg, and Nicolas M Thiéry. Directed nonabelian sandpile models on trees. *Communications in Mathematical Physics*, 335:1065–1098, 2015.
- [5] Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality: An explanation of the $1/f$ noise. *Physical Review Letters*, 59(4):381, 1987.
- [6] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.
- [7] Norman Biggs. Algebraic potential theory on graphs. *Bulletin of the London Mathematical Society*, 29(6):641–682, 1997.
- [8] Norman L. Biggs. Chip-firing and the critical group of a graph. *Journal of Algebraic Combinatorics*, 9:25–45, 1999.
- [9] Lucie Bréchet, Denis Brunet, Gwénaél Birot, Rolf Gruetter, Christoph M. Michel, and João Jorge. Capturing the spatiotemporal dynamics of self-generated, task-initiated thoughts with EEG and fMRI. *Neuroimage*, 194:82–92, 2019.

- [10] Verena Brodbeck, Alena Kuhn, Frederic von Wegner, Astrid Morzelewski, Enzo Tagliacuzzi, Sergey Borisov, Christoph M. Michel, and Helmut Laufs. EEG microstates of wakefulness and NREM sleep. *Neuroimage*, 62(3):2129–2139, 2012.
- [11] Rafael C. Carrasco and José Oncina. Learning stochastic regular grammars by means of a state merging method. In Rafael C. Carrasco and José Oncina, editors, *Grammatical Inference and Applications*, pages 139–152. Springer Verlag, 1994.
- [12] Yao-ban Chan, Jean-François Marckert, and Thomas Selig. A natural stochastic extension of the sandpile model on a graph. *Journal of Combinatorial Theory, Series A*, 120(7):1913–1928, 2013.
- [13] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4):359–394, 1999.
- [14] Alexander Clark and Franck Thollard. PAC-learnability of probabilistic deterministic finite state automata. *Journal of Machine Learning Research*, 5:473–497, 2004.
- [15] Matthew S. Crouse, Robert D. Nowak, and Richard G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 46(4):886–902, 1998.
- [16] James P. Crutchfield. Observing complexity and the complexity of observation. In Harald Atmanspacher and Gerhard J. Dalenoort, editors, *Inside Versus Outside*, pages 235–272. Springer Verlag, 1994.
- [17] James P. Crutchfield and Karl Young. Inferring statistical complexity. *Physical Review Letters*, 63(2):105–108, 1989.
- [18] Arnaud Dartois, Francesca Fiorenzi, and Paolo Francini. Sandpile group on the graph d_n of the dihedral group. *European Journal of Combinatorics*, 24(7):815–824, 2003.
- [19] Colin de la Higuera and Jose Oncina. Learning stochastic finite automata. In Georgios Paliouras and Yasubumi Sakakibara, editors, *Grammatical Inference: Algorithms and Applications*, pages 175–186. Springer Verlag, 2004.
- [20] Hanna Derets and Chrystopher L. Nehaniv. The study of the transformation semi-group of the Abelian and directed non-Abelian sandpiles. In D. Marc Kilgour, Herb Kunze, Roman Makarov, Roderick Melnik, and Xu Wang, editors, *Recent Developments in Mathematical, Statistical and Computational Sciences (AMMCS 2023)*, Springer Proceedings in Mathematics & Statistics. Springer Nature, in press.

- [21] Deepak Dhar. Theoretical studies of self-organized criticality. *Physica A: Statistical Mechanics and its Applications*, 369(1):29–70, 2006. Fundamental Problems in Statistical Physics.
- [22] Pál Dömösi and Chrystopher L. Nehaniv. *Algebraic Theory of Automata Networks: An Introduction*, volume 11 of *Monographs in Discrete Mathematics and Applications*. Society for Industrial and Applied Mathematics (SIAM), 2005.
- [23] Attila Egri-Nagy, James D. Mitchell, and Chrystopher L. Nehaniv. SgpDec: Cascade (de)compositions of finite transformation semigroups and permutation groups. In Hoon Hong and Chee Yap, editors, *Mathematical Software – ICMS 2014*, pages 75–82. Springer Verlag, 2014.
- [24] Attila Egri-Nagy, James D. Mitchell, and Chrystopher L. Nehaniv. SgpDec: Semi-group Decomposition. <https://github.com/gap-packages/sgpdec>, 2014. Accessed: September 10, 2024.
- [25] Samuel Eilenberg. *Automata, Languages, and Machines. Volume B*. Pure and Applied Mathematics. Academic Press, New York, 1976.
- [26] Matthias Gärtner, Verena Brodbeck, Helmut Laufs, and Gaby Schneider. A stochastic model for EEG microstate sequence analysis. *Neuroimage*, 104:199–208, 2015.
- [27] The GAP group. GAP – Groups, Algorithms, and Programming. <https://www.gap-system.org>. Version 4.13.1; 2024, Accessed: September 10, 2024.
- [28] Markus Gschwind, Christoph M. Michel, and Dimitri Van De Ville. Long-range dependencies make the difference — Comment on “A stochastic model for EEG microstate sequence analysis” by Gärtner et.al. *NeuroImage*, 117:449–455, 2015.
- [29] Brian Jacobson, Andrew Niedermaier, and Victor Reiner. Critical groups for complete multipartite graphs and cartesian products of complete graphs. *Journal of Graph Theory*, 44(3):231–250, 2003.
- [30] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.
- [31] Arjun Khanna, Alvaro Pascual-Leone, Christoph M. Michel, and Faranak Farzan. Microstates in resting-state EEG: Current status and future directions. *Neuroscience Biobehavioral Reviews*, 49:105–113, 2015.

- [32] Seungki Kim and Yuntao Wang. A stochastic variant of the Abelian sandpile model. *Journal of Statistical Physics*, 178(3):711–724, 2020.
- [33] Jochen Kindler, Daniela Hubl, WK Strik, Thomas Dierks, and Thomas König. Resting-state EEG in schizophrenia: Auditory verbal hallucinations are related to shortening of specific microstates. *Clinical Neurophysiology*, 122(6):1179–1182, 2011.
- [34] Caroline J. Klivans. *The mathematics of chip-firing*. CRC Press, 2018.
- [35] Thomas Koenig and Daniel Brandeis. Inappropriate assumptions about EEG state changes and their impact on the quantification of EEG state dynamics. *Neuroimage*, 125:1104–1106, 2016.
- [36] Thomas Koenig, Dietrich Lehmann, Marco C.G. Merlo, Kieko Kochi, Daniel Hell, and Martha Koukkou. A deviant EEG brain microstate in acute, neuroleptic-naïve schizophrenics at rest. *European Archives of Psychiatry and Clinical Neuroscience*, 249(4):205–211, 1999.
- [37] Thomas Koenig, Leslie Prichep, Dietrich Lehmann, Pedro Valdes Sosa, Elisabeth Braeker, Horst Kleinlogel, Robert Isenhardt, and E. Roy John. Millisecond by millisecond, year by year: Normative EEG microstates and developmental stages. *Neuroimage*, 16(1):41–48, 2002.
- [38] Kenneth Krohn and John Rhodes. Complexity of finite semigroups. *Annals of Mathematics*, 88(1):128–160, 1968.
- [39] Hyeon-Kyu Lee and J. H. Kim. An HMM-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973, 1999.
- [40] Dietrich Lehmann, Pascal L. Faber, Silvana Galderisi, Werner M. Herrmann, Toshihiko Kinoshita, Martha Koukkou, Armida Mucci, Roberto D. Pascual-Marqui, Naomi Saito, Jiri Wackermann, Georg Winterer, and Thomas Koenig. EEG microstate duration and syntax in acute, medication-naïve, first-episode schizophrenia: a multi-center study. *Psychiatry Research: Neuroimaging*, 138(2):141–156, 2005.
- [41] Haipeng Lian, Yingjie Li, and Yunxia Li. Altered EEG microstate dynamics in mild cognitive impairment and Alzheimer’s disease. *Clinical Neurophysiology*, 132(11):2861–2869, 2021.
- [42] Patricia Milz. KeyPy: The KEY EEG analysis toolbox. Zenodo. doi: 10.5281. 2015.

- [43] Patricia Milz, Pascal L. Faber, Dietrich Lehmann, Thomas Koenig, Kieko Kochi, and Roberto D. Pascual-Marqui. The functional significance of EEG microstates — Associations with modalities of thinking. *Neuroimage*, 125:643–656, 2016.
- [44] Cristopher Moore and Martin Nilsson. The computational complexity of sandpiles. *Journal of Statistical Physics*, 96(1):205–224, 1999.
- [45] Chrystopher L. Nehaniv and Elena Antonova. Simulating and reconstructing neurodynamics with epsilon-automata applied to electroencephalography (EEG) microstate sequences. In *IEEE Symposium on Computational Intelligence, Cognitive Algorithms (SSCI), Mind, and Brain*, pages 1753–1761. IEEE Press, 2017.
- [46] Chrystopher L. Nehaniv, Elena Antonova, David G. Haydock, Hanna Derets, Sebastián Dohnány, Zixuan Gao, Reinoud Maex, Shabnam Kadir, and Robert Leech. Novel computational methods for predicting transitions in spatiotemporal neurodynamics between attention and mind-wandering. Final Scientific Report to the United States Air Force Office of Scientific Research (AFOSR), Award number FA9550-19-1-7034, 2023.
- [47] Chrystopher L. Nehaniv. Complexity of finite aperiodic semigroups and star-free languages. In J. Almeida, G. Gomes, and P. Silva, editors, *Semigroups, Automata, Languages*, pages 195–209. World Scientific, 1996.
- [48] Keiichiro Nishida, Yosuke Morishima, Masafumi Yoshimura, Toshiaki Isotani, Satoshi Irisawa, Kay Jann, Thomas Dierks, Werner Strik, Toshihiko Kinoshita, and Thomas Koenig. EEG microstates associated with salience and frontoparietal networks in frontotemporal dementia, schizophrenia and Alzheimer’s disease. *Clinical Neurophysiology*, 124(6):1106–1114, 2013.
- [49] Nick Palmer and Paul W. Goldberg. PAC-learnability of probabilistic deterministic finite state automata in terms of variation distance. *Theoretical Computer Science*, 387(1):18–31, 2007.
- [50] Nicolás Perry and P. M. Binder. Finite statistical complexity for sofic systems. *Physical Review E*, 60(1):459, 1999.
- [51] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [52] Frank Redig. Course 14 - Mathematical aspects of the Abelian sandpile model. In Anton Bovier, François Dunlop, Aernout van Enter, Frank den Hollander, and Jean

- Dalibard, editors, *Mathematical Statistical Physics*, volume 83 of *Les Houches*, pages 657–729. Elsevier, 2006.
- [53] John L. Rhodes, Kenneth B. Krohn, and Bret R. Tilson. *Algebraic Theory of Machines, Languages, and Semigroups*. Academic Press, 1968. Editor: Michael A. Arbib.
- [54] Dana Ron, Yoram Singer, and Naftali Tishby. On the learnability and usage of acyclic probabilistic finite automata. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, pages 31–40, 1995.
- [55] Felix Schlegel, Dietrich Lehmann, Pascal L. Faber, Patricia Milz, and Lorena R. R. Gianotti. EEG microstates during resting represent personality differences. *Brain Topography*, 25(1):20–26, 2012.
- [56] Cosma Shalizi, Kristina Klinkner, and James Crutchfield. CSSR: Causal state splitting reconstruction. <https://github.com/stites/CSSR>, 2015. Accessed: September 10, 2024.
- [57] Cosma Rohilla Shalizi and Kristina Lisa Klinkner. Blind construction of optimal nonlinear recursive predictors for discrete sequences. In Max Chickering and Joseph Y. Halpern, editors, *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference (UAI 2004)*, pages 504–511, Arlington, Virginia, 2004. AUAI Press.
- [58] Cosma Rohilla Shalizi, Kristina Lisa Shalizi, and James P. Crutchfield. Pattern discovery in time series, Part I: Theory, algorithm, analysis, and convergence. *Journal of Machine Learning Research*, pages 02–10, 2002.
- [59] Chihiro Shibata and Ryo Yoshinaka. Marginalizing out transition probabilities for several subclasses of PFAs. In Jeffrey Heinz, Colin Higuera, and Tim Oates, editors, *Proceedings of the Eleventh International Conference on Grammatical Inference*, volume 21 of *Proceedings of Machine Learning Research*, pages 259–263, University of Maryland, College Park, MD, USA, 05–08 Sep 2012. PMLR.
- [60] Valeria B. Strelets, Pascal L. Faber, J. Golikova, Vladimir Y. Novototsky-Vlasov, Thomas Koenig, Lorena R. R. Gianotti, John H. Gruzelier, and Dietrich Lehmann. Chronic schizophrenics with positive symptomatology have shortened EEG microstate durations. *Clinical Neurophysiology*, 114(11):2043–2051, 2003.
- [61] Christopher C. Strelhoff and James P. Crutchfield. Bayesian structural inference for hidden processes. *Physical Review E*, 89(4):042119, 2014.

- [62] Luke Tait, Francesco Tamagnini, George Stothart, Edoardo Barvas, Chiara Monaldini, Roberto Frusciante, Mirco Volpini, Susanna Guttman, Elizabeth Coulthard, Jon T. Brown, Nina Kazanina, and Marc Goodfellow. EEG microstate complexity for aiding early diagnosis of Alzheimer’s disease. *Scientific Reports*, 10(1):17627, 2020.
- [63] Gábor Tardos. Polynomial bound for a chip firing game on graphs. *SIAM Journal on Discrete Mathematics*, 1(3):397–398, 1988.
- [64] Bret Tilson. Chapter XII: Complexity of Semigroups and Morphisms. In Samuel Eilenberg, editor, *Automata, Languages and Machines. Volume B*, volume 59B of *Pure and Applied Mathematics*, pages 313–385. Academic Press, New York, 1976.
- [65] Evelin Christiana Toumpakari. *On the Abelian Sandpile Model*. Phd dissertation, The University of Chicago, Chicago, Illinois, 2005.
- [66] Jan Van Den Heuvel. Algorithmic aspects of a chip-firing game. *Combinatorics, Probability and Computing*, 10(6):505–529, 2001.
- [67] Sicco Verwer, Rémi Eyraud, and Colin De La Higuera. PAUTOMAC: A probabilistic automata and hidden Markov models learning competition. *Machine Learning*, 96:129–154, 2014.
- [68] Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.
- [69] Frederic von Wegner, Enzo Tagliazucchi, and Helmut Laufs. Information-theoretical analysis of resting state EEG microstate sequences—non-Markovianity, non-stationarity and periodicities. *Neuroimage*, 158:99–111, 2017.
- [70] Milena C. Wiemers, Helmut Laufs, and Frederic von Wegner. Frequency analysis of EEG microstate sequences in wakefulness and NREM sleep. *Brain Topography*, 37(2):312–328, 2024.
- [71] Gale Young and Aiston S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22, 1938.