ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

GUILHERME GRÄF SCHÜLER

GRAPHS OF GROWTH:
DETECTING INFANT MOVEMENT ANOMALIES WITH GRAPH CONVOLUTIONAL NETWORKS

Porto Alegre
2024

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

GRAPHS OF GROWTH:

DETECTING INFANT MOVEMENT ANOMALIES WITH GRAPH CONVOLUTIONAL NETWORKS

GUILHERME GRÄF SCHÜLER

Master Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Prof. Dr. Márcio Sarroglia Pinho

Porto Alegre

2024

Ficha Catalográfica

S415g Schüler, Guilherme Gräf

Graphs of growth : detecting infant movement anomalies with graph convolutional networks / Guilherme Gräf Schüler. – 2024.
105 p.
Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Márcio Sarroglia Pinho.

1. Deep learning. 2. General movements assessment. 3. Graph convolutional networks. I. Pinho, Márcio Sarroglia. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

Guilherme Gräf Schüler

Graphs of growth: detecting infant movement anomalies with graph convolutional networks

This Master Thesis has been submitted in partial fulfillment of the requirements for the degree of Master of Computer Science, of the Graduate Program in Computer Science, School of Technology of the Pontifícia Universidade Católica do Rio Grande do Sul.

Sanctioned on April 09, 2024.

COMMITTEE MEMBERS:

Prof. Dr. Kayvan Najarian (University of Michigan)

Prof. Dr. Lucas Silveira Kupssinskü (PPGCC/PUCRS)

Prof. Dr. Márcio Sarroglia Pinho (PPGCC/PUCRS - Advisor)

ACKNOWLEDGEMENTS

I am immensely grateful for Professor Márcio Pinho's encouragement and guidance throughout my studies. It was thanks to his patience and support that I was able to properly adapt and enjoy this experience as much as I had.

To my parents, I thank them for their unconditional support. Both have been a reliable spring of liveliness, assurance, and safety for the last two years, which I consider to be invaluable resources for life.

Vitória, I am happy that you have decided to spend your time with me, as every moment we spend together is bliss, and it has not been different for these past years. Wherever we are, I know that, with you, moving ahead leads to light and happiness.

For inspiring me in my academic journey from Philosophy to anywhere else, I am unmeasurably thankful to Rogério Severo. Writing, which takes up such a large part of this, is for me a pleasure due to you. Although less frequent, talking to you remains a great source of joy.

I also thank my friends and professors who supported me with or without their intention. Finally, thanks to CAPES and the PPGCC of the Pontifical University of Rio Grande do Sul for allowing me to undergo this course with exclusive commitment.

GRAFOS DO CRESCIMENTO: DETECTANDO ANOMALIAS EM MOVIMENTOS INFANTIS COM REDES CONVOLUTIVAS

RESUMO

Transtornos do desenvolvimento cognitivo (TDC) é uma designação geral para deficiências decorrentes do mau desenvolvimento do sistema nervoso. Bebês prematuros são a população mais afetada e, embora não haja cura para TDCs, tratamentos estão disponíveis assim que o transtorno é identificado. A Avaliação de Movimentos Gerais (AMG) é uma ferramenta de diagnóstico para discernir entre neurodesenvolvimento típico e indicativo de risco em bebês abaixo de 6 meses de idade via a observação de repertórios de movimento específicos – alguns dos quais são anormais e atribuem risco à criança. Apesar de seu alto valor preditivo para CDDs, a AMG é pouco utilizada em ambientes clínicos devido a um programa de treinamento e certificação complexo e custoso. O objetivo desta dissertação é desenvolver uma metodologia para a automatização da AMG: de registros em vídeo do movimento de bebês em ambientes hospitalares, para a classificação de movimento normal e anormal e posterior identificação de risco. Foi desenvolvido um sistema de classificação baseado em Redes Neurais Convolutivas de Grafo para atribuir risco ou não-risco de CDDs em três datasets publicamente disponíveis, contendo sequências com dados posicionais de bebês. No total, dados de 137 bebês foram usados para treinar o algoritmo de classificação. Mudanças à arquitetura interna da rede e etapas de regularização foram feitas a fim de adaptá-la ao caráter ruidoso dos dados. Conduzimos um processo de otimização de hiperparâmetros em diversas configurações experimentais, submetendo nosso modelo a diferentes tipos de dados, tanto intra-datasets – treinamento e teste no mesmo dataset – quanto inter-datasets.

Palavras-Chave: Avaliação de Movimentos Gerais, redes neurais convolutivas, aprendizado profundo, movimentos gerais.

GRAPHS OF GROWTH: DETECTING INFANT MOVEMENT ANOMALIES WITH GRAPH CONVOLUTIONAL NETWORKS

ABSTRACT

Cognitive development disorder (CDD) is an umbrella term for impairments arising from the maldevelopment of the nervous system. Premature infants are the most affected population and although most CDDs have no cure, treatment is available as soon as the disorder is identified. The General Movements Assessment (GMA) is a diagnostic tool for discerning between typical and disorder-like neurodevelopment of infants below 6 months of age via the observation of specific movement repertoires -- some of which are abnormal and attribute risk to the infant. Despite its high predictive value for CDDs, GMA is scarcely used in clinical settings due to a difficult and costly training and certification program. This dissertation's purpose is to develop a methodology for automating GMA: from video-recordings of moving infants in hospital settings to the classification of normal and abnormal movement and later risk identification. We developed a classification system based on a Graph Convolutional Neural Network to sort out infant skeleton time-series data of three different publicly available datasets into risk of CDDs and no-risk of CDDs. In total, data from 137 infants were used to train our classification algorithm. Changes to the internal architecture of the network and regularization steps were made to adapt to the noisy nature of our data. We performed hyperparameter optimization on different experimental setups, subjecting our model to different data, both intra-datasets – training and testing on the same dataset and – and inter-datasets.

Keywords: General Movements Assessment, convolutional neural networks, deep learning, general movements.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AF	Abnormal fidgety movement
AUC-ROC	Area under the Receiver Operating Characteristic curve
BCE	Binary cross-entropy
BN	Batch normalization
BSID	Bayley Scales of Infant and Toddler Development
CA	Corrected age
CDD	Cognitive development disorders
CH	Chaotic movement
ChA	Chronological age
CI	Confidence interval
CNN	Convolutional neural network
CP	Cerebral palsy
CS	Cramped-synchronized
CUS	Cranial ultrasound
DL	Deep learning
DNN	Deep neural network
DT	Decision tree
FCNN	Fully-connected neural network
FFT	Fast-Fourier transform
FM	Fidgety movement
FM-	Absent fidgety movement
FN	False negative
FP	False positive
FPS	Frame per second
GAN	Generative adversarial network
GA	Gestational age
GBDT	Gradient boosting decision tree
GCN	Graph convolutional network
GD	Gradient descent
GM	General movement
GMA	General Movements Assessment
GMDS	Griffith Mental Development Scales
GMT	General Movements Toolbox
GMT	General Movements Trust
GNB	Gaussian naïve Bayes
HINE	Hammersmith Infant Neurological Examination
IMS	Inertial measurement system
kNN	k-Nearest neighbors
LDA	Linear discriminant analysis
LDOF	Large displacement optical flow
LOOCV	Leave-one-out cross-validation
LR	Logistic regression
LSTM	Long short-term memory
MLP	Multi-layer perceptron
MRI	Magnetic resonance scan
MSE	Mean squared error
MS-STGCN	Multi-stage Spatio-temporal graph convolutional network
NAN	Not a number
NPV	Negative prediction value
PCA	Principal component analysis
PCHIP	Piecewise cubic Hermite interpolating polynomial

PLSR	Partial least squares regression
PMA	Post-menstrual age
PPV	Positive prediction value
PR	Poor repertoire movement
PR-AUC	Area under the Precision-Recall curve
PT	Postterm
RF	Random forest
SMOTE	Synthetic minority over-sampling techniques
VAE	Variational Autoencoder

SUMMARY

1 INTRODUCTION

Cognitive development disorder (CDD) is an umbrella term for impairments arising from the maldevelopment of the nervous system. Premature infants, above all, are the most affected population. In very preterm groups — born 8 or fewer weeks before a normal birth —, the likelihood of developmental delay reaches 47% (Caesar et al., 2021). Overall, the degree of prematurity is associated with an increased risk of motor, behavioral, and cognitive impairments (Craciunoiu & Holsti, 2017). Although most CDDs have no cure, treatments are available — and most effective when started early in the patient's life.

For treatments to be addressed, diagnosis must happen. Many methods have been developed to evaluate the neurodevelopment of newborns. Among them, three are of the most predictive value regarding developmental outcomes (Craciunoiu & Holsti, 2017; Novak et al., 2017). The Test of Infant Motor Performance (TIMP), the Hammersmith Infant Neurological Examination (HINE), and the General Movements Assessment (GMA) all aim to assess an infant's motor functions such as posture, movement patterns, and reflexes. While HINE yields better outcome predictions when applied in infants between 2 and 24 months of age, the TIMP and GMA achieve better results in infants under 4 and 5 months, respectively. In this work, special interest will be given to the GMA.

Briefly, the General Movements Assessment (GMA) is a reliable method for discerning between typical and atypical neurodevelopment of infants in the first 5 months of life. It is especially valuable in indicating risk of cerebral palsy (CP), as its implementation in the clinical setting has dropped CP's age of diagnostic from 19.5 to 9.5, on average (Maitre et al., 2020). GMA is defined as a standardized, non-invasive, comfortable, and cheap method (Einspieler et al., 2004). It is feasible for use in neonatal intensive care units and requires no effort from the infant. Two experts lead the assessment — usually pediatricians or neurologists —, analyzing the spontaneous movement of an infant for 5 to 10 minutes. The infant must lie in a supine position in the incubator, bed, or mattress, depending on its age and health state (see Figure 1). The target movements, *general movements*, should occur naturally, given comfortable temperature and clothing conditions (Einspieler et al., 1997). Age-specific movement patterns can be labeled as normal or abnormal. Multiple assessments during the development of the infant are brought together and assessed, so intervention can take place. Although abnormal outcomes do not indicate specific disorders, they flag a degree of risk for their development.

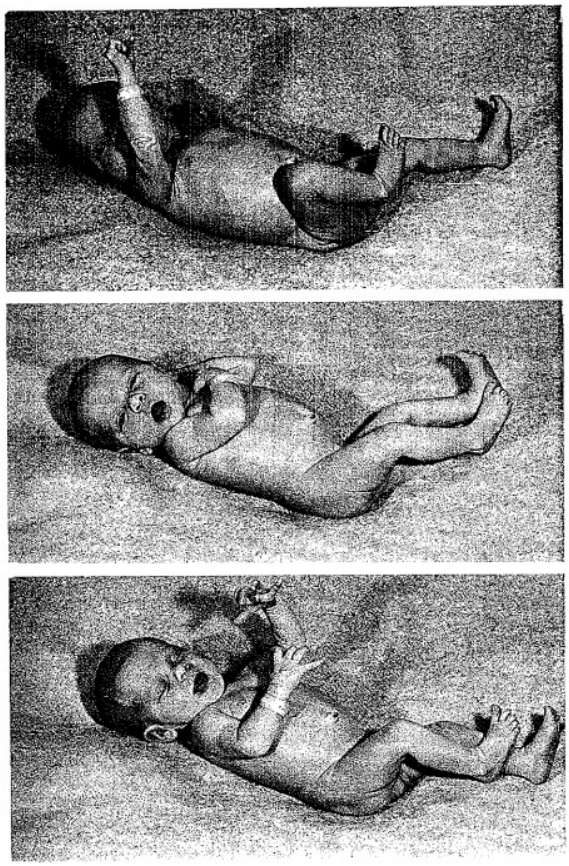


Figure 1 - Spontaneous movements from a newborn in the supine position.
(Prechtl, 1977).

A unique characteristic of the GMA method, when compared to its alternatives, is that the assessment procedure involves little interaction between the assessor and the infant. Whereas HINE and TIMP assessments rely on information such as response to handling and visual/auditory cues, GMA relies solely on the observation of involuntary infant movements (Haataja et al., 1999; Kim et al., 2011; Romeo et al., 2016). As a result, recordings of the infant's movement are sufficient, and even desirable, for the assessment. In fact, GMA is widely applied non-concurrently via pre-recorded footage (Einspieler et al., 1997).

Additionally, while being far easier to conduct than most of its alternatives, GMA has not been established as a clinical routine standard — for GMA can only be performed by certified experts. To certify oneself, proper training, regular practice, and recalibration are needed, which is not always available or affordable. Certifications are issued by the General Movements Trust

(GMT)[1] and require fulfilling the basic course. Both basic and advanced courses are available via the GMT and vary from US$800,00 to US$1000,00. The GMA's manual can be bought for approximately US$55,00[2]. Ultimately, this renders a predictive tool inaccessible and hard to apply widely.

With that in mind, attempts at automating this diagnostic tool have been proposed — largely within the computer vision, machine, and deep learning literature (Schmidt et al., 2019; Silva et al., 2021). GMA is of special interest to automation attempts precisely because of its non-intrusive method: the only material needed for prediction is the uninterrupted movement of the infant. With proper preparation of the infant and setup, infant movement should be captured, and GMA automation can be applied to the data afterward. Movement may be captured either by sensors directly attached to the infant's body – such as accelerometers, gyroscopes, and marker-based motion capture (Gao et al., 2019; Meinecke et al., 2006) –, or indirectly via algorithms applied to video footage. As a result, data often contains time-series of acceleration, orientation, or coordinates of specific parts of the infant's body. The second step is classification (Groos, Adde, Aubert, et al., 2022; Ni et al., 2023). Specifically, to answer the question: how can this data be used to successfully classify movement patterns as normal or abnormal?

Both steps make up challenges. Regarding data capture, most difficulties arise from the unique set of shapes and motions that newborns' and infants' bodies possess, as well as the limited space in which to attach sensors (Groos, Adde, Stoen, et al., 2022; Hesse et al., 2020; Li et al., 2021). Although different approaches for transforming raw data will be discussed in this work, our focus is on classification. Data capture system's literature is broad, even if limited to infants' bodies, and discussing it will be left out. Whenever mentioned, however, further explanation will be given.

---

[1] https://general-movements-trust.info/.

[2] Price data was accessed on February 7, 2024. Different institutions have different prices. Two institutions were accessed for this work, corresponding to the latest provided courses. The Karolinska Institutet basic course — offered on March January 24/25th, 2024 —, is priced at US$935,00. Advanced courses often cost an additional fee of around US$900,00, and can only be taken by having completed the basic course. Different courses, their dates and costs, were accessed via the GMA Trust website (https://general-movements-trust.info/47/dates).

For this specific task, classification is binary: movement pattern is either normal or abnormal. To achieve proper classification, a relevant set of quantitative features must be extracted from the raw data. These might come in distinct ways: hand-made features, thought of specifically for their power to represent certain movement patterns (i.e. the entropy of the right arm), general statistics derived from movement (i.e. the standard deviation of each limb's velocity), or embeddings computed by deep learning algorithms – which often don't carry a semantical interpretation (Balta et al., 2022; Ni et al., 2023; Redd et al., 2021; Tong et al., 2022). Once decided upon, the resulting classifiers are evaluated on cohorts of healthy and unhealthy infants, and their performance is reported. Medical cohorts, however, are often particular to certain research groups and cannot be shared between different studies for comparison. This yields a lack of benchmark data in which classifiers can be commonly assessed, and hinder discussion between researchers (Silva et al., 2021). Recent work has tried to establish publicly available datasets (Gong et al., 2022; K. McCay et al., 2022; K. D. McCay et al., 2019; Tong et al., 2022); however, these are small in size and greatly imbalanced – being prone to the overfitting of classifiers and lack of generalization.

1.1 Objectives

This work aims to tackle some of these issues by developing a classifier and designing different experimental setups using publicly available datasets. Our overall objective is to provide a methodology for the preprocessing of data from different sources and its classification using deep neural networks. In particular, we adapt a Graph Convolutional Network to train on labeled data provided from three datasets: the MINI-RGBD, RVI-38, and PMI-GMA. Graphs can be powerful representational tools, especially when dealing with skeleton-like data. By representing 2D coordinates of different joints as nodes, and connecting as edges, a graph become a natural representation of infant movement data.

By training and validating our model on different data partitions from distinct data, a more robust classifier can be achieved. Additionally, these experimental setups might serve as paradigms for further testing and comparison of different classification systems in the literature, and promote a clearer discussion of the benefits and detriments of existing methods. That said, our specific goals in this work are:

- Define a preprocessing pipeline for the time-series pose data of three publicly available datasets that allows its conjoined use as training and testing data;
- Adapt a Graph Convolutional Network to perform action recognition – where the action is general movements –, and subsequently classify entire sequences as normal or abnormal patterns of movement;
- Design different experimental setups for the training, validation, and testing, of the deep learning model;
- Adjust the model's inner structure and hyperparameters to best perform on the available data.

1.2 Structure

This dissertation is organized as follows. The next section contains the theoretical framework, which is divided into two subjects. First, general movements and GMA's methodology and reliability will be tackled, which will familiarize the reader with the classification problem and serve as the rationale for specific choices regarding our classifier. Secondly, deep learning architectures, computing modules, and metrics used for reporting performance – focused on Graph Convolutional Networks – will be described. Related work on state-of-the-art approaches to GMA automation is discussed in Section 3. The methodology for the collection and preparation of data, as well as the development of the classifier and experiments, will be presented in Section 4. Section 5 contains the results concerning the model's performance on different experiments and different parameters. Finally, a discussion of our results and considerations for future work are contained in Section 6.

2 THEORETICAL FRAMEWORK

This section describes the basic concepts which will be used throughout this work. First, an overview of the General Movements Assessment will be conducted, going over its objective, procedure, and reliability within clinical practice. Then, the core architecture of a deep learning model will be presented. Computing modules, basic operations, regularization techniques, and the metrics used for reporting performance in this work will be outlined.

2.1 General Movements Assessment (GMA)

The General Movements Assessment (GMA) is a diagnostic tool in the field of neonatal neurology. It is used for the early detection and prediction of neurodevelopmental outcomes in newborns and infants. The assessment involves observing infants' spontaneous movements and evaluating their quality and maturity. Understanding these movements and the procedure by which they are evaluated is essential if any extension of the tool is desired.

This section goes through an overview of the General Movements Assessment, its objective, development, techniques, procedures, and application in clinical practice. A comparison between GMA and its alternatives, such as the TIMP and the HINE, is also included.

2.1.1 Objective and development

The nervous system is responsible for coordinating many of the body's functions, from sensory perception and movement to memory and learning (Ludwig et al., 2022). Taking care of it should start early and be a priority. Neurological examination of newborns – whether pre-term or full-term – is of utmost importance. Notably, infants who may have suffered birth or pregnancy trauma, and those who were born into families with a complicated medical history, should all be subject to some kind of assessment. These are often called "at risk" infants (Prechtl, 1977).

The important purpose of the neurological examination is to document the newborn's neurodevelopment. If irregular, follow-up care is extremely beneficial (Novak et al., 2017; Sokołów et al., 2020). If an examination predicts motor maldevelopment, for instance, early physiotherapy may be crucial for the infant's subsequent years (Sant et al., 2021). When there is a risk of intellectual disability, programs tend to focus on education and support for caregivers (Hadders-Algra, 2021).

The main hypothesis that enabled GMA as a reliable method for neurological examination is that low-risk and high-risk infants have different spontaneous motor patterns (Einspieler et al., 2004). Subsequent investigation confirmed the validity of this hypothesis, as clinicians were first unable to discern a quantitative discrepancy until the introduction of video recording, which revealed a qualitative contrast (Ferrari et al., 1990).

Specifically, a subset of movements, called "general movements", had contrasting patterns depending on the development of the newborn's brain (Prechtl, 1990). The GMA's method, then, relies on the qualitative observation of such movements to predict the neurodevelopmental outcome of infants.

2.1.2 What are general movements?

Infants show a variety of spontaneous movement patterns (see Figure 1), which are motor activities not related to external stimulation (de Vries et al., 1982; Prechtl, 1990). General movements (GMs) are a prominent subset of an infant's spontaneous movements, characterized generally as being of special complexity (Prechtl, 2001). These movements are observed already during the fetal stage, in infants with 9 weeks postmenstrual[3] age (Einspieler et al., 1997). The same movement pattern continues until the end of the second month of corrected age[4] (CA) when new GM patterns gradually appear.

Broadly, general movements have been defined as gross movements that involve the whole body, lasting from a few seconds to a minute. They "wax and wane in intensity, force, and speed, and their onset and end are gradual" (Prechtl, 1990). Most of the "extension or flexion of arms and legs is complex, with superimposed rotations and often slight changes in direction of the movement" (*ibid.*). Across the years, general movements have been consistently characterized by predicates such as "always graceful in character" (de Vries et al., 1982). In fact, being "fluent and elegant" is part of its common definition (Prechtl, 1990), which continued to be used in

[3] Ascribing neonate age is standardized. "Gestational age" refers to the time elapsed between the first day of the mother's last menstrual period and the day of delivery. "Chronological age" is the time elapsed since birth. "Postmenstrual age" is defined as the sum of gestational and chronological age (Committee on Fetus and Newborn, 2004).

[4] "Corrected age" is a term often used to describe children up to 3 years of age who were born preterm. It is computed as the chronological age reduced by the number of months born before expected day of delivery (40 weeks). This terminology will be used throughout this work.

subsequent research (Bos et al., 1997; Einspieler et al., 1997; Ferrari et al., 1990; Prechtl, 1990) and in the recent literature (Gima et al., 2019; Y.-C. Wu et al., 2021).

More relevant to the General Movement Assessment is the description of age-specific patterns. Healthy infants from term age to their second month show GMs called *writhing movements* (Einspieler et al., 1997). It is important to note that, despite having different names, fetal/preterm movements, and writhing movements have a very similar appearance (Einspieler & Prechtl, 2005). Figure 2 illustrates the chronological order of GMs.

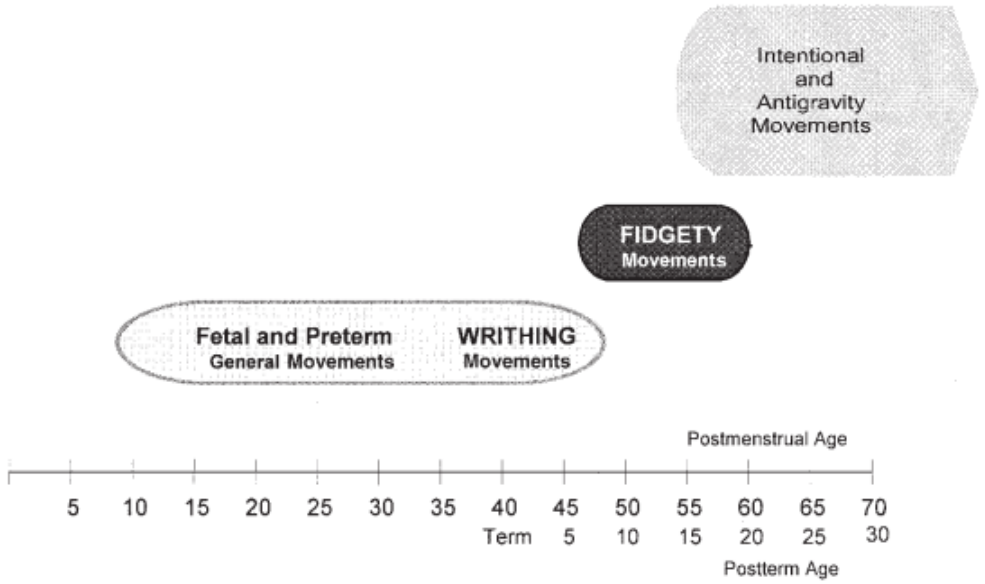


Figure 2 - Developmental course of general movements.

(Einspieler & Prechtl, 2005).

Writhing movements have small to moderate amplitude and slow to moderate speed. Extensor movements, such as an extension of the arms and legs, are common during this period. These movements are typically of an elliptical – writhing – form. Furthermore, co-contraction of antagonist muscles is frequent (Prechtl & Hopkins, 1986).

At the age of 6 to 9 weeks post-term, writhing movements gradually transform into another pattern, named *fidgety movements*. Fidgety movements have a circular shape, small amplitude, moderate speed, and irregular acceleration of the neck, trunk, and limbs (Einspieler et al., 1997). They are continuous when the infant is awake and not in focused attention. Present until 15 to at most 20 weeks corrected age (see Footnote 3), initially occurring as isolated

events, then increasing in frequency, and finally wearing off to be replaced by antigravity and intentional movements (Einspieler et al., 2016).

When these patterns do not occur in the way they are supposed to, they are labeled as *abnormal* movements and indicate a risk of cognitive and motor maldevelopment. For the writhing period, three abnormalities can occur:

- **Poor repertoire (PR):** sequences of successive movement that are monotonous and do not happen in the complex way that GMs normally have (Ferrari et al., 1990);
- **Cramped-synchronized (CS):** GMs appear rigid while limb and trunk muscles contract and relax almost in a synchronized manner (Einspieler et al., 1997)
- **Chaotic (Ch):** all limb's movements are of large amplitude occurring in a chaotic and abrupt manner, with neither fluency nor smoothness (*ibid*.).

An infant (B) with poor-repertoire GMs, born at 28 weeks postmenstrual age, is contrasted to a healthy infant (A), born at term age, in Figure 3. Poor-repertoire GMs are recognizable by the nearly identical frames.

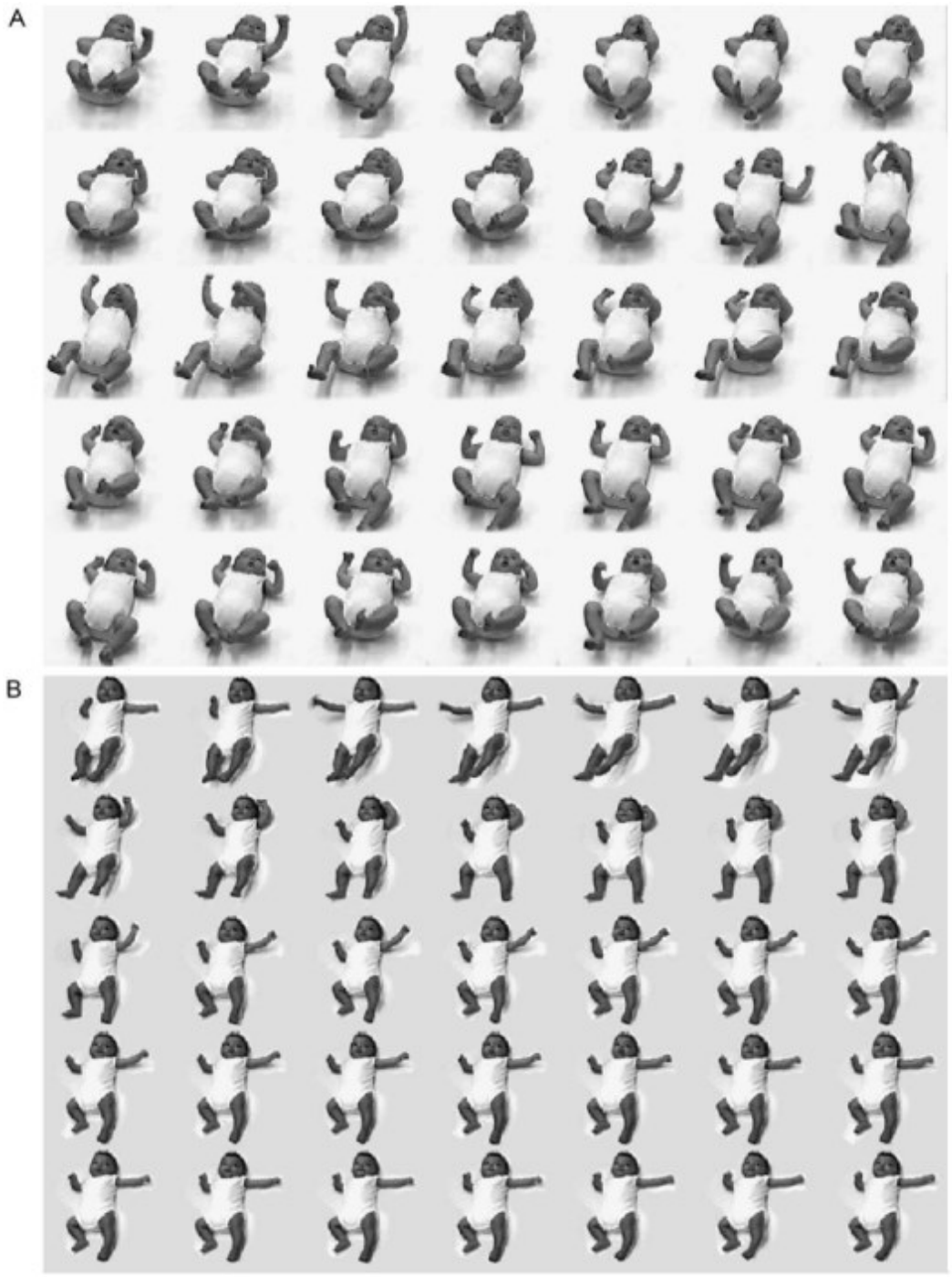


Figure 3 - Video frames (left to right, top to bottom) of recordings from infants in the fidgety period.

Infant A (above) has healthy GMs, infant B (below) has abnormal, PR GMs. Interval between frames is 0.24s (Hadders-Algra, 2004).

Regarding the fidgety period, patterns are said to be abnormal when either: fidgety movements are never observed from ages 6 to 20 weeks post-term; or they have exaggerated amplitude, speed, and jerkiness. Such patterns are called “absent” and “abnormal” (Prechtl et al., 1997). A summary of normal and abnormal GMs is given in Table 1.

Table 1 - Definition of GMs and their abnormalities, in accordance with the General Movements Trust. *(Prechtl, 2001).*

Age Period	Normal GMs	Abnormal GMs
Prenatal and preterm	Whole body movements. Variable sequence of arm, leg, neck, and trunk motion. Gradual beginning and end, with irregular intensity, force, and speed. Complex extension of arms and legs. Superimposed rotations and change of direction.	*Poor repertoire* (PR): monotonous sequence of successive movements. Motion is not as complex as seen in normal GMs. *Cramped-synchronized* (CS): rigid movements that lack smoothness. Limb and trunk muscles contract and relax simultaneously. *Chaotic* (Ch): Large amplitude, no fluency nor smoothness, and abrupt.
Term age until 8 weeks' post-term age (writhing period)	Small-to-moderate amplitude, slow-to-moderate speed. Fast and large extension motion, especially in the arms. Elliptical in shape. Co-contraction of antagonist muscles.	
6 to 20 weeks' postterm age (fidgety period)	Circular movements of small amplitude, moderate speed, variable neck, trunk, and limbs acceleration. Continual in the awake infant, except during focused attention periods. Initially happen as isolated events, gradually increasing in frequency and finally wearing off to be replaced by intentional movements.	*Absent* (FM-): Fidgety movements are never observed from ages 6 to 20 weeks post-term. *Abnormal* (AF): moderately or greatly exaggerated amplitude, speed, and jerkiness.

2.1.3 Techniques and procedure

The procedure to assess motor function by GMA is agreed upon. This subsection goes over it briefly, to the extent that it contributes to this work. Concerning the setup, recording footage is usually preferred over *in loco* assessment, allowing re-playability and slow-motion features, and avoiding interference of observers in the infant's behavior (Einspieler et al., 2004). The best view of the baby is obtained by filming from above (*ibid.*). Besides, the infant should lie in the supine position in the incubator, bed, or on a mattress, depending on its age, and temperature should be comfortable, as well as the infant's clothing, so that movement is not limited (Einspieler et al., 1997).

Video recording and analysis of GMs should be done longitudinally. One hour of recording is sufficient so that smaller intervals (3-5 minutes) containing GMs are identified and separated from the full footage. In the end, a full collection of an infant's movement recording should contain (i) many recordings of the preterm period; (ii) one recording of term age; (iii) one recording between 3 and 6 weeks; and (iv) at least one recording during fidgety movement period (Einspieler et al., 1997). Several assessments done longitudinally are often called "developmental trajectories," describing the quality of general movements across an infant's

first 2 months of age. Table 2 shows the developmental trajectories for 6 distinct infants, all born preterm (<37 weeks), and their neurological outcome at 2 years of age.

Table 2 - Developmental trajectories and longitudinal GM assessments.
•, *moment of birth; N, normal movement or outcome; DR, developmental retardation; CP, cerebral palsy. Adapted from (Einspieler et al., 1997).*

	Postmenstrual age (weeks)																
Infant nº	*32*	*33*	*34*	*35*	*36*	*37*	*38*	*39*	*40*	*41*	*42*	*43*	*44*	*45*	*46*	*47*	**2y**
1			•		N	N	N		N			N					**N**
2					•	PR		PR		PR			PR		N		**N**
3			•		CS		PR		PR			PR			PR		**DR**
4	•	PR	Ch		Ch		Ch		PR		PR			PR			**CP**
5			•	PR	PR	CS		CS			CS				CS		**CP**
6	•		CS	CS	CS	CS		CS			CS				CS		**CP**

These are valuable to the extent that similar trajectories might yield similar developmental outcomes. Further analysis, especially of abnormal movements, can be done via scoring standards. Scoring standards are irrelevant to this proposal's topic, namely, computer-based approaches for automating GMA. Similarly, procedures for interscorer agreement and examiner knowledge prior to assessment are equally out of scope. However, it is noteworthy that diagnosis done via GMA is always subjected to interscorer agreement analysis, and is only considered reliable with high scores. For scoring standards, interscorer agreements, and examiner's prior knowledge requisites, see respectively (Einspieler et al., 1997; Ferrari et al., 1990; Prechtl, 1977), (Peyton et al., 2021), and (Y.-C. Wu et al., 2021).

2.1.4 GMA in clinical practice: prediction, advantages, and reliability

Accumulated evidence supports GMA as a reliable tool for indicating neuromotor risk, particularly mild cognitive delays, and cerebral palsy (Akcakaya et al., 2019; Bosanquet et al., 2013; Caesar et al., 2021; Craciunoiu & Holsti, 2017; Kwong et al., 2018; Noble & Boyd, 2012; Novak et al., 2017). Furthermore, GMA can consistently indicate the risk of specific cerebral palsy (CP) types. Spastic and dyskinetic CP are the most common subtypes, with over 70% prevalence in CP populations (Novak et al., 2017). These are characterized by a combination of loosened and stiffened musculature, which results in poor walking and reflexes, feeding and posture issues, and comorbidities such as musculoskeletal, behavioral, and intellectual

problems (*ibid.*). Cramped-synchronized GMs seem to be highly correlated with the late development of severe spastic CP (Einspieler & Prechtl, 2005). Additionally, the risk of both spastic and dyskinetic variations of CP is accurately indicated by the absence of fidgety movements (FM-) (*ibid.*).

Overall, GMA is used in clinical settings for the early identification of developmental disorders and the assessment of infants at risk. The earlier the identification of risk, the better intervention plans can be suitably planned for the infant and its family. Historically, cerebral palsy is accurately diagnosed in infants between 12 to 24 months of age (Te Velde et al., 2021). Implementing GMA as a clinical routine dropped this mark, on average, from 19.5 (95% CI [16.2 - 22.8]) to 9.5 months of age (95% CI [4.5 – 14.6]) (Maitre et al., 2020).

Metrics for the reliability of GMA and other neurological assessments are computed by comparing the predicted outcome and the subsequent, decisive, outcome. This final outcome is usually acquired via neurological follow-ups based on scales, such as the Griffith Mental Development Scales (GMDS) and the Bayley Scales of Infant and Toddler Development (BSID). Briefly, these scales measure motor, cognitive, language, and social-emotional skills of the infant and output a score, that is representative of the infant's neurodevelopment (Balasundaram & Avulakunta, 2022; Pino et al., 2022). This enables standardized comparison between different methods.

Although choosing an assessment technique should consider the primary purpose of examination, GMA has consistently high values for positive and negative prediction, sensitivity, and specificity. In a systematic review of 8 neonatal assessments of preterm infants up to 4 months, GMA achieved the best scores for all metrics mentioned, followed by the Test of Infant Motor Performance (Noble & Boyd, 2012). In a 19 studies review (Bosanquet et al., 2013), GMA had estimates of 95 to 100% sensitivity and 96 to 98% specificity in 2 studies. In another 2 studies made during the fidgety period, assessment on preterm infants achieved sensitivity and specificity of 87 to 100% and 82% to 95%, respectively (*ibid.*). Four additional studies comprising 326 infants with a 29% prevalence of CP had a pooled sensitivity/specificity of 98% (95% CI [73-100]) and 91% (95% CI [83-95]) respectively (*ibid.*). Magnetic resonance scans (MRIs) and Cranial ultrasounds (CUSs) had similar results in predicting CP, with 86-100% sensitivity and 87-97% specificity (Bosanquet et al., 2013). A review of 6 systematic reviews and 2 evidence-based guidelines described MRI and GMA as the most reliable tools for predicting

CP before 5 months CA, with 86-89% and 98% sensitivity scores respectively (Novak et al., 2017). Conducted in a clinic where GMA is standardly used, a study with 80 infants (60% prevalence of motor disabilities) achieved 95.8% sensitivity and 87.5% (Akcakaya et al., 2019).

Even though the General Movement Assessment is well supported, there are two main disadvantages. Firstly, abnormal movement patterns indicate risk at different degrees of reliability. The absence of fidgety, as seen before, is highly predictive (see Table 3 for a summary of predictive scores for different patterns). One drawback is that specificity values for GMA of *writhing* movements are low (59% CI [45-71]) (Kwong et al., 2018). This indicates an elevated number of false positives, which means that a high number of infants might be wrongly induced into intervention. Secondly, since general movements stop manifesting after 20 weeks corrected age, GMA can no longer be applied (Caesar et al., 2021).

Table 3 - Summary of predictive scores for distinct abnormal patterns by GMA.
Adapted from (Kwong et al., 2018).

Test	Sensitivity, % (95% CI)	Specificity, % (95% CI)	PPV range, %	NPV range, %
Cramped-synchronized	70 (54-82)	97 (74-100)	36-100	74-94
Poor-repertoire	93 (86-96)	59 (45-71)	8-68	80-100
Absent fidgety	89 (66-97)	81 (64-91)	6-56	96-100

Despite all, GMA is often judged for its subjective analysis and evaluation. GMA is consistent and reliable when carried out by experts, but falls off when applied by practitioners with a basic training course and fewer years of experience (Y.-C. Wu et al., 2021). Requiring vast experience and advanced training damages GMA's applicability and hampers it as a standard in clinical practice. This motivates automation attempts, to which we now turn.

2.2 Automating GMA with Deep Neural Networks

Deep neural networks (DNNs) are a class of machine learning algorithms based on neurons – computational units responsible for mathematical operations – organized in hierarchical layers. By continuously passing information through these layers, and adjusting themselves, DNNs can extract useful features for different machine learning tasks, such as image segmentation and action recognition (Bishop, 2006; Theodoridis, 2015). When the goal is human action recognition, a particular type of DNN architecture, namely Graph Convolutional Networks

(GCNs), has been thoroughly used and shown great results due to its power to convolute on graph-structure data (Ahmad et al., 2021; S. Zhang et al., 2019).

Data from infant movement can be easily represented in graphs (see Figure 4) both spatially and temporally, and GCNs offer the computational modules required for dealing with these data.

Figure 4 - A skeleton-like structure superimposed on a video recording frame of a moving infant.

The points can be naturally modelled as nodes whose edges are the connecting lines (Groos, Adde, Stoen, et al., 2022).

Specifically, we will be dealing with time-series of 2D skeleton data, i.e., 2D coordinates for different body parts over time. Skeleton data naturally translates to graphs, and can be further modeled temporally. That said, this section briefly goes over the core functioning of the generalized DNN structure, the convolution operation and its use in Convolutional Neural Networks (CNNs), and its extension to graph input data in GCNs. Additionally, we describe regularization techniques and common metrics used for evaluating the performance of these DL systems, which will be later used in our performance report.

2.2.1 Deep neural networks and learning

Deep neural networks owe their origin to the foundational work of Frank Rosenblatt in 1957 with the perceptron, a rudimentary linear classifier (Rosenblatt, 1957). Although the perceptron

originally represented a single-layer network, we use the term here as a precursor to understanding more complex architectures, which usually involve several, *deep*, layers. The architecture of a perceptron encompasses three fundamental parts: input, weighted summation, and output. These layers play distinctive roles in processing information for various machine-learning tasks, such as classification, regression, and segmentation (Georgevici & Terblanche, 2019).

Perceptrons can be stacked parallelly to create a Multi-layer perceptron (MLP), known for its power to capture non-linear relationships in the data (Murtagh, 1991). MLPs share the essential features of a perceptron, with the addition of an activation function after each layer, from which non-linearity emerges. In it, the input units x_i are connected to the second layer's units h_j, called a *hidden layer* through weights $w_{i,\mathrm{j}}$. For a hidden unit h_1, its input will be the weighted sum of the input units x_i with the weights $w_{i,1}$, and similarly for all hidden units. Then, the output of the weighted sum is non-linearly transformed by an activation function σ.[5] Thus, the output from a hidden unit h_j is

$$h_j = \sigma\left(\mathbf{w}_{\mathrm{j}}\,\mathbf{x}\right) \tag{1}$$

where w_{j} is the vector of weights going from the input vector $\mathbf{x}$ to the hidden unit of index j. The activation function σ can take many forms, but it is here taken to be the sigmoid function (Dubey et al., 2022), for clarity purposes. The sigmoid function simply reduces its input into the range $[0,1]$, and allows for efficient training in later phases – as will be seen. Finally, the output layer can be mapped by ϕ to whatever range of values corresponds to the desired outcome. It is similar to Equation 1 in which

$$\hat{y}_k = \phi(w_{\mathrm{k}}\,\boldsymbol{h}) \tag{2}$$

[5] Our description of neural networks omit biases, as they are unessential in this introduction. Similarly to weights, biases are parameters usually *added* to each previous layer state, such that a hidden layer h_j is computed as the product of the previous layer, e.g. the input, with the corresponding weights *summed* with a set of input-independent biases. During the learning phase both weights and biases are adjusted, but we will consider weights solely.

where w_k is the vector of weights going from the hidden layer $\boldsymbol{h}$ to the output of index k (our example has only one output). The final activation function, ϕ, may or may not differ from the activation function of the hidden layers, σ. Well-known non-linear functions include *softmax* for multi-class classification (Dubey et al., 2022) and ReLU, which will be further described in Section 4. Figure 5 illustrates the MLP architecture. Stack more than three hidden layers with, for instance, 64 units each, and the MLP is commonly denoted as a *deep* neural network (Goodfellow et al., 2016). If all units – from the input, hidden layers, and output – are connected, then the resulting architecture is often called a fully-connected neural network (FCNN) and belongs to the most basic class of DNNs.

Learning, however, plays no role in the process of going from input to output, the *feed-forward* or forward pass step of a neural network. Opposite to that, learning is present in the *backward pass,* in which the output is compared to the expected value, e.g., the image label, and weights are adjusted. In the next paragraphs, we will quickly go over the process of computing loss functions and updating weights through the backpropagation function, as these are the essential features of every neural network learning capabilities.

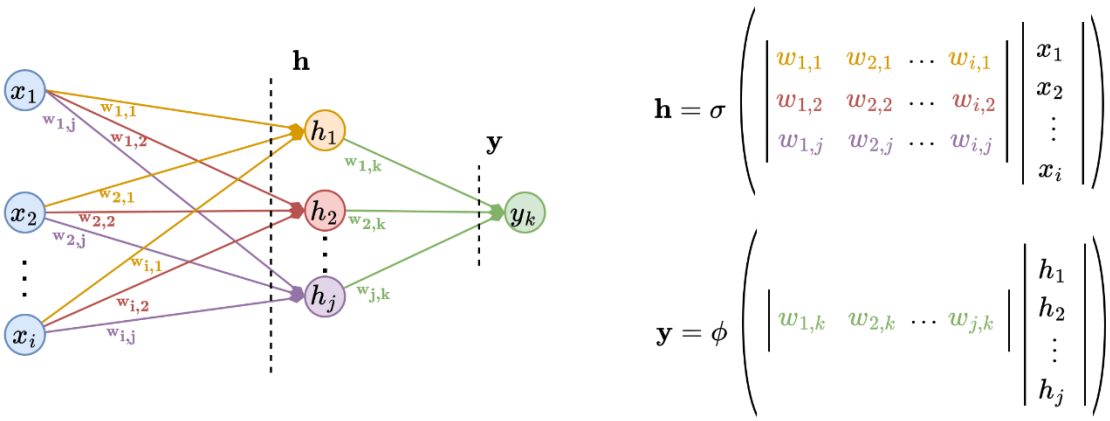


Figure 5 - Multi-layer perceptron's basic architecture.

The example illustrates an input of i elements and one hidden layer $\boldsymbol{h}$ with j hidden units. The computation for $\boldsymbol{h}$ and the output $\boldsymbol{y}$ in matrix form is also shown, wherein weights and layer units are multiplied and transformed by their respective activation functions.

Say our toy example has as its input an image that contain a toddler (positive class). Consider our MLP model computed an output of 0.4, corresponding to the probability p that the input

belongs to a positive true label. We would like to say our model made a bad prediction, but most importantly that it requires some adjustments. The adjustment of our model takes place in the learning phase – and requires computing a loss function and using it to update its weights.

Loss functions, or cost functions, provide an "error measure" by which a model can guide its learning. Different functions exist, each with their own advantages and disadvantages (Wang et al., 2022). For binary classification problems, binary cross-entropy (BCE) is a common choice. Given one example as input, BCE computes the negative log of the probability of the true label, y, given the predicted outcome $\hat{y}$, as seen in Equation 3. Forward passes, however, often happen in *batches* instead of single examples, where a defined number of examples is passed through the model before updating its parameters. That said, a more general way of presenting BCE is as the average of the negative log-likelihood of n examples (Janocha & Czarnecki, 2017):

$$BCE = \frac{1}{n}\left(\sum_{i=1}^{n} y^{(i)} \times log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \times log\left(1 - \hat{y}^{(i)}\right)\right) \quad (3)$$

Large loss values indicate the model needs large changes to better perform. To guide *how* change must happen in a model's parameters, we can compute the effect of tweaking particular parameters on the loss function. Most often, backpropagation is the algorithm responsible for indicating in which direction and with what magnitude each of the model's parameters needs to be adjusted to minimize the loss function (Goodfellow et al., 2016). Its detailed inner workings are complex and will not be covered here.

Once backpropagation yields *gradient* values for each parameter – broadly corresponding to their tweaking effect on cost values --, an optimization algorithm such as gradient descent does the model's update (Nielsen, 2015). Gradient descent (GD) can be separated into three variants: batch, stochastic, and mini-batches gradient descent. These are related to the way the training data is allocated and offer a trade-off between accuracy and efficiency for updating its parameters $\boldsymbol{\theta}$ (Ruder, 2017). Broady, batch GD updates $\boldsymbol{\theta}$ after a full pass through the network; stochastic GD (SGD) performs updates after *each* randomly selected training example; and mini-batch GD updates for every mini-batch of n training examples. A full training cycle for a model usually defines an optimization algorithm, such as SGD, together with a **learning rate** α

hyperparameter[6]. If working with mini-batches, **batch size** also becomes a hyperparameter. For optimization to happen, a cost function (or functions) must be specified – usually reflecting the task the model aims to complete. Thus, a single forward-pass, backpropagation, and parameter updates over all training data establish an **epoch**. Typically, training happens for several epochs until the model's performance stabilizes or reaches a satisfactory level, and the exact number of epochs is a hyperparameter.

This subsection aimed to familiarize the reader with how deep networks emerge from computations between weights and activation functions. It also described the basic learning mechanisms shared by most complex DNNs. The next subsections go over the convolution operation, its use in DL architectures, and graph inputs in GCNs.

2.2.2 Convolutional neural networks

The convolution operation enables images (and any other two-dimensional data, e.g. time series) to be fed as 2D vectors and to extract meaningful features from their local regions (O'Shea & Nash, 2015). It works by sliding a filter, or **kernel**, over the input data – in this case, a two-dimensional image –, and computing a combination of the kernel elements and the image pixels, often called a **feature map**. Whereas FCNNs had connections going from every neuron in a layer to every neuron in the following layer, CNNs do not. This is due to kernels being smaller than the input matrix, and combining fewer elements into their computations. For instance, Figure 6 shows the computation of a 2×2 kernel in the top-left region of a 4×4 input, which connects to only four elements of the input – instead of being fully-connected to all 16 of them. This property is called *sparse connectivity* and amounts to a great part of the representational capacity of CNNs, as well as their efficiency (Alzubaidi et al., 2021; Goodfellow et al., 2016). Secondly, CNNs use the same parameters more than once for a single kernel "slide": the same four parameters of our 2×2 kernel will be convoluted to different regions of the input. *Parameter sharing* – the second important property of CNNs – favors the learning of features that well-represent spatial patterns and structures.

[6] To distinguish between values *of* the model – such as weights and biases – and values that influence its learning/training process, the term hyperparameter is henceforth used (as is usual among the literature (Goodfellow et al., 2016; Nielsen, 2015; Ruder, 2017)).

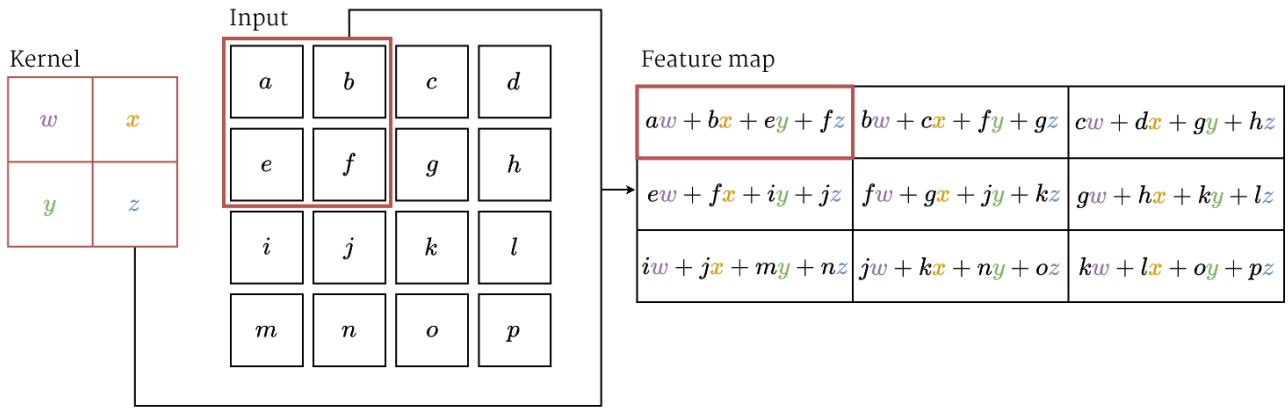


Figure 6 - 2D convolution.

One must imagine the 2×2 kernel sliding through the 4×4 matrix, from the top-left to the bottom-right corner. In this example, the convolution operation happens only where the kernel lies entirely within the input tensor. Adapted from (Goodfellow et al., 2016).

Convolution is implemented in CNNs as a convolutional layer, which yields a number of feature maps corresponding to different kernels working on the input. Feature maps' elements are the central learnable parameters of CNNs, which will be updated throughout backward passes. During the implementation of CNNs, **kernel size**, **stride size**, the number of **feature maps**, and other values, are considered hyperparameters to be set. Stride size corresponds to the number of elements (e.g., pixels) by which the kernel is slid every step. Typically, the output from convolutional layers is run through a non-linear activation function prior to being modified by a *pooling layer*, which further enhances the generalizability and invariance of the net.

Pooling is responsible for summarizing the output of the previous layer among its neighbors. Summary statistics such as the maximum and mean value of s elements help to further make features invariant to small translation changes in previous layers (see Figure 7). Similarly to convolution, pooling is applied considering a kernel and stride size, and different choices for these hyperparameters affect the granularity of selected features and the number of parameters.

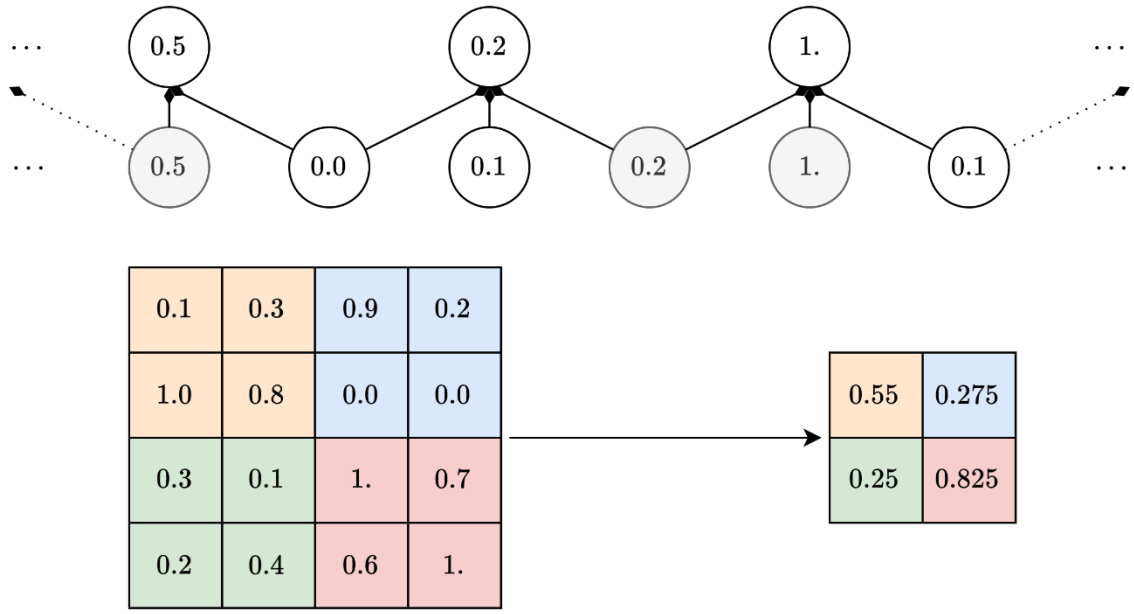


Figure 7 - Pooling operation on the output of non-linear activation functions (post-convolution).

Top: *Max pooling with width three and stride of one element operating on a 1D vector.* ***Bottom:*** *Average pooling with a kernel size of* 2×2 *and stride of two elements (vertically and horizontally) on a 2D vector. Adapted from (Goodfellow et al., 2016).*

Wide kernels for pooling will likely lead to coarser summaries of the data – which might represent general patterns of the data –, while smaller kernels might capture finer details, but are susceptible to capturing noise and ungeneralizable patterns (O'Shea & Nash, 2015).

A convolutional neural network specializes in grid-structured data. Most exceedingly, it deals with images and two-dimensional data. All modules in a CNN are designed to capture spatial patterns and to generalize on them while trying not to be misdirected by noise or information from specific regions of the input. As the data of interest to this dissertation is *video* data, CNNs seem especially promising in dealing with video *frames*, but might still not be the best way to represent *human motion*. The next section goes over graph convolutional networks, an extension of CNNs that allows graph-structured data to be used as its input.

2.2.3 Graph convolutional networks

CNNs mostly rely on grid patterns for feature extraction –, and once *non*-Euclidean structures are considered CNNs lose most of their advantages (S. Zhang et al., 2019). A prime example of a non-Euclidean structure is the graph, where entities and their relationships can be represented by nodes and edges – which carry additional information by themselves. Intuitively,

graphs are great representations for human motion. Mapping to the GMA domain, consider different body parts represented as nodes and their connecting bones as edges, and graphs turn into a useful structure for representing human poses – which in turn encapsulates the needed information for identifying movements. In fact, graphs became widely used for portraying skeleton-based data, with an emerging class of DNNs' architectures adapted CNNs to work on its non-Euclidean structure (Ahmad et al., 2021; Gori et al., 2005).

Graph convolutional networks (GCNs) allow the basic operations of CNNs to work on graph-structured inputs and differ mostly in the way they aggregate information. As illustrated in Figure 8, grid-like topologies such as images allow us to presume a fixed number of neighboring elements and an order-like structure, while graphs do not. Convolution and pooling, thus, must be adapted for this context. Briefly, we next describe the graph structure and the fundamental functions of GCNs.

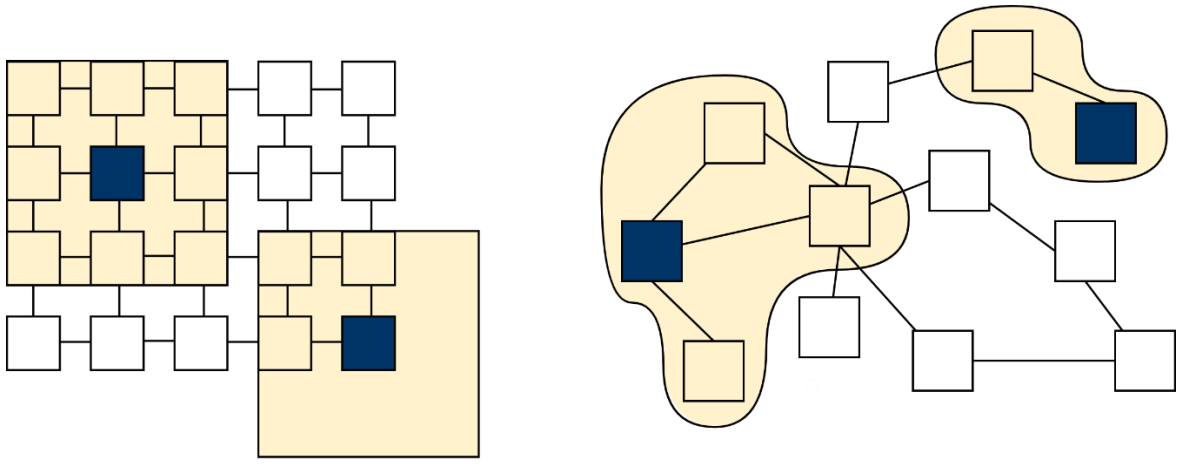


Figure 8 - Range of operations on different topologies.

*(**Left**) on a two-dimensional Euclidean grid topology and (**right**) on a graph topology. Blue squares represent the center node on which the kernel, in yellow, operates.*

Generally, the input of a GCN will be a graph $G = (V, E, \boldsymbol{X}_V, \boldsymbol{X}_E)$, where $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{(i,j) \mid i \text{ is connected to } j\}$ are the set of nodes and edges, respectively; $\boldsymbol{x}_i$ is the feature vector of node v_i, and $\boldsymbol{X}_V = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ is the n-sized set of feature vectors of all nodes. Similarly, $\boldsymbol{x}_{(i,j)}$ denotes the feature vector of the edge (i,j) and $\boldsymbol{X}_E = \{\boldsymbol{x}_{(i,j)} | (i,j) \in E\}$ is the m-sized set of feature vectors of all edges. Connections of the graph are contained in an adjacency graph $\mathbf{A} \in \mathbb{R}^{n \times n}$ where $\mathbf{A}_{(i,j)} = 1$ if nodes i and j are connected, otherwise 0. As with any DNN,

distinct tasks can be undertaken by GCNs, e.g., classification, segmentation, and regression. Moreover, these tasks can be aimed at specific nodes, edges, or entire graphs (Zhou et al., 2020). Graphs may be construed in numerous ways, and a discussion of how GCNs can adapt to each one is beyond the scope of this work. Let us limit ourselves, then, to undirected and unweighted graphs – where $\boldsymbol{X}_V$ exhausts the total number of features – and to graph-level classification, where an entire graph structure is classified.

The first step of a GCN is usually a convolution layer, same as with CNNs. The required information for a convolution layer is the set of features $\boldsymbol{X}_V$ and the graph's adjacency matrix $\mathbf{A}$, and its output consists of a new graph with updated *embedding* vectors. Node-wise, convolution is often called *message-passing* (Ward et al., 2022) as it collects information from neighboring nodes to update its own features and a set of learnable parameters, mathematically defined as

$$\mathbf{h}_v^{(s)} = \sigma\left(\sum_{j \in \boldsymbol{\mathcal{N}}(v)} \frac{\mathbf{h}_j^{(s-1)}\mathbf{w}}{\sqrt{|\boldsymbol{\mathcal{N}}(v)||\boldsymbol{\mathcal{N}}(j)|}}\right) \tag{4}$$

where $\mathbf{h}_v^{(s)}$ is the embedding vector of node v at step (s), σ is an activation function, $\boldsymbol{\mathcal{N}}(v)$ and $\boldsymbol{\mathcal{N}}(j)$ are the neighbors of nodes v and j, $\mathbf{h}_j^{(s-1)}$ is the embedding vector of node j at step $(s-1)$, and $\mathbf{w}$ is a learnable weight matrix. As with CNNs, learnable parameters $\mathbf{w}$ are shared among node updates, ensuring the benefits of parameter-sharing previously mentioned. Overall, Equation 4 sets the value of a node embedding to be the weighted sum of its direct neighboring nodes' embeddings, normalized by a value representing the total number of "potential" message transmitters available to the target node (Kipf & Welling, 2017). A full pass through a convolutional layer, thus, yields n embedding vectors with unchanged connections (i.e., $\mathbf{A}$ is unaltered). Stacking these layers allows nodes to integrate information from far-reaching nodes since their neighbor's embeddings will also contain information from *their* neighboring nodes. In fact, most GCN architectures rely on the stacking of convolutional layers to capture potential relationships across the entire graph (X.-M. Zhang et al., 2021).

Node-wise embeddings, however, are not sufficient for *graph*-level classification. To achieve this, it is common to further aggregate these embeddings into a single vector. This corresponds to the pooling operation in traditional CNNs and is most often achieved by

compressing the node embeddings via mean- or max-pooling, and using the resulting vector $\mathbf{z}$ as input to an MLP or FCNN, which can finally generate an appropriate output (Zhou et al., 2020). Figure 9 concisely illustrates a single pass through a basic GCN for graph-level classification.

This subsection exhausts the architectural design concepts of deep neural networks which will be used in this work. To ensure these models perform well on actual data, and that they generalize well to unseen information, regularization techniques are indispensable The next section goes over these techniques.

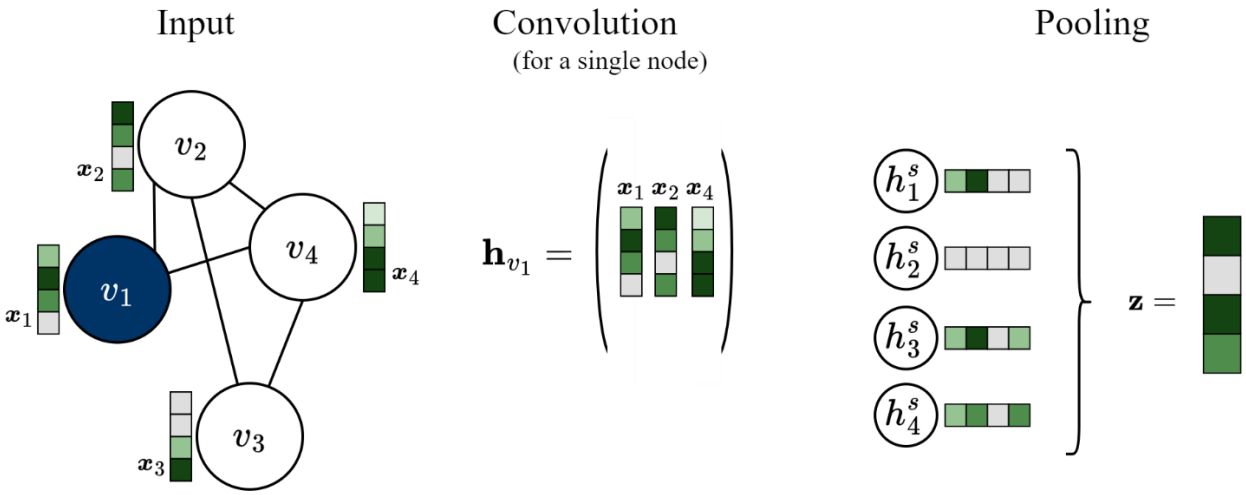


Figure 9 - Overview of operations in a GCN architecture.

From left to right, the schematic illustrates the convoluting of node v_1 and its neighbors' features to form $\boldsymbol{h}_{v_1}$. After repeating convolution for all nodes, max-pooling of the final node embeddings is computed to create a graph-level embedding, $\boldsymbol{z}$.

2.2.4 Regularization techniques

When training DNNs – especially with small quantities of data –, learning can sometimes be hampered by noisy, un-generalizable data. To keep models from "memorizing" these faulty patterns, two common methods have been widely used: batch normalization and dropout. Recall the notion of a loss value when training models. If the loss decreases during training but increases on unseen data (during testing), it is probable that the model is *overfitting*. That is, it has possibly learned noisy patterns that do not generalize to unseen data (Kukačka et al., 2017). Regularization can be seen as a general term for all techniques whose aim is to reduce overfitting (Tian & Zhang, 2022).

When training, each layer's learnable parameters get updated for every batch of data. After each update, the parameters' distribution is mostly random, which hinders a fast and optimal convergence and subsequent learning (Tian & Zhang, 2022). Batch normalization (BN) standardizes the activation output of every batch or mini-batch. Implemented as a layer in a DNN, it computes the mean and standard deviation of its input, separately for each feature dimension. Then, BN normalizes distributions based on these statistics, which yields a more stable and generalizable model. This, in turn, encourages the model to learn more robust features of the data and avoid overfitting (Goodfellow et al., 2016).

Dropout is also commonly implemented as a layer and does a quite simple job. A dropout layer of $p = 0.5$ put after a fully connected layer of 64 neurons means that each of these neurons have a 50% chance of being set to 0. Besides reducing the number of trained parameters, dropout prevents co-adaptation of specific neurons and improves the chance that learned features are global and generalizable (Tian & Zhang, 2022).

2.2.5 Metrics for ML model evaluation

To standardly evaluate the reliability of machine learning models we need standard metrics. This subsection goes over some of the most often used metrics and some of the usual steps for evaluating models. Since infants' motion data is usually small-sized, the most used method for training and validation of models is Leave-One-Out cross-validation (LOOCV) (Adde et al., 2013; K. D. McCay et al., 2019). This consists of separating data into N sets where one set is used for validation and $N - 1$ is used for training. This can be done for n times, where n is the number of subsets the data was split into (Bishop, 2006). Then, the metrics can be computed as the average between all splits made via LOOCV.

Sensitivity and specificity were already mentioned in section 2.1.4 but will be described here. Let us say that in our classification problem, two discrete outcomes are possible: normal movement (0) and abnormal movement $(+1)$. Consider that the "positive" class is $+1$. In this case, *sensitivity* (or recall) is the proportion of positive classes that were successfully classified, i.e., abnormal movements that were correctly classified as abnormal movements. *Specificity* (Spc.) tells us the proportion of negative classes that were correctly classified. For instance, a sensitivity value of 1 means that all of the abnormal movements in the data were correctly classified as such; a specificity value of 0 means that all of the normal movements were

mistakenly classified as abnormal (Trevethan, 2017). The number of positive classes successfully classified among all positive classes is called the positive predictive value (PPV). The same holds for negative values, and it is called negative predictive value (NPV) (*ibid.*). See Figure 10 and Equations 5-8 for further reference.

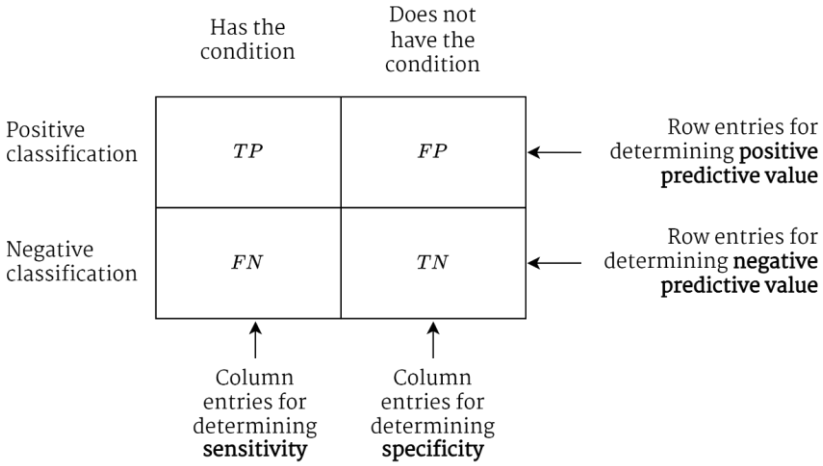


Figure 10 - Reference diagram for sensitivity, specificity, PPV, and NPV.

Adapted from (Trevethan, 2017).

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{Specificity} = \frac{TN}{FP + TN} \tag{6}$$

$$\text{PPV} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{NPV} = \frac{TN}{FN + TN} \tag{8}$$

Accuracy and AUC-ROC are also frequent in statistical analysis for ML. Accuracy is simply the ratio between correct predictions and the total number of predictions (Bishop, 2006). The Area Under the Curve of the Receiver Operating Characteristic (AUC-ROC) is a common metric to summarize a model performance. Put briefly, ROC is a probability curve that plots sensitivity against one-minus-specificity at different thresholds, corresponding to decision boundaries to classify instances as either positive or negative (Hajian-Tilaki, 2013).

By computing the area under the probability curve, we generate the AUC (see Figure 11), a summary measure representing a model's ability to distinguish between positive and negative classes. An AUC of 1 means that the model correctly distinguishes between all positive and negative classes; an AUC of 0 means that it predicts all positives as negatives and vice-versa.

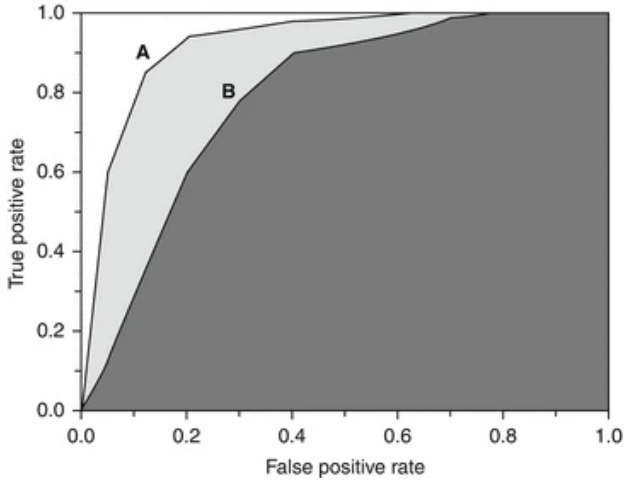


Figure 11 - AUC-ROC for two classifiers, A and B.

In this example, A has a larger AUC value than B, corresponding to the light grey area (Melo, 2013).

When dealing with unbalanced datasets, where negative classes outnumber positive, summary statistics such as the F1-score and the PR-AUC are better suited. The F1 score is the harmonic mean of PPV and sensitivity (also called precision and recall, respectively). The PR is a curve plotting precision and recall at different thresholds, and the PR-AUC is a summary value describing the overall ability of a model to classify negative and positive classes in a highly unbalanced set of data.

3 RELATED WORK

This section contains a summary of related works found through a systematic literature review. A brief explanation of the review's methodology, followed by its main findings and summary tables containing relevant information from selected studies will be presented along with a small discussion.

This review was conducted using the PRISMA guidelines (Page et al., 2021), firstly carried out on May 24, 2023, and updated on January 13, 2024, on Embase[7], Pubmed[8], Scopus[9], Web of Science[10], IEEE Xplore[11], and ACM Digital Library's Guide to Computing Literature[12]. Table 4 shows the queried terms, which could be contained in the study's title, abstract, or keywords.[13] Only studies from 2012 onwards were included (11-year period).

Table 4 - Literature review's search terms separated by topic (measurement, movement, and population).

Search item	Search terms
(i)	'ai' OR 'features' OR 'computer-based' OR 'video**' OR 'sensor*' OR 'automat*' OR 'accelerometer*' OR 'inertial measurement unit' OR 'imu' OR 'motion analysis' OR 'instrumented' OR 'deep*learning' OR 'machine*
(ii)	'general movement* assessment' OR 'gma' OR 'spontaneous movement*' OR 'fidgety movement*' OR 'writhing movement*'
(iii)	'infant*' OR 'newborn*' OR 'child*' OR 'preterm' OR 'neonate*' OR 'neonatal' OR 'cerebral palsy' OR 'high-risk'

Eligibility criteria for inclusion are listed in Table 5, and studies had to meet all criteria to be included. Additionally, systematic reviews and case studies involving only one subject were excluded. A total of 1042 articles were collected across the six queried databases. After duplicate removal and a first screening, 93 articles remained – which were fully read and

[7] https://embase.com
[8] https://pubmed.ncbi.nlm.nih.gov
[9] https://scopus.com
[10] https://webofknowledge.com
[11] https://ieeexplore.ieee.org
[12] https://dl.acm.org/
[13] See Table 35 in the APPENDIX for full query strings for each database.

included. Additionally, three studies were added via manual search. As a result, 38 papers were included in this systematic review.

Table 5 - Literature review's inclusion eligibility criteria and their description.

Inclusion criteria	Description
Population	Infant population of ≤ 6 months corrected age (CA);
Quantitative measurement	Employ quantitative instruments for measuring infant movement;
Comparison with GMA-related measurements	Contain analysis of the quantitative movement data concerning GM or derived-diagnosis (e.g., CDDs, CP), be it from statistical analysis or ML;
Outcome	Report outcome measures for classification/multivariate analysis;
Additional	Written in English; peer-reviewed; full articles.

The methodology for this review is illustrated in Figure 12. Table 6 and Table 7 describe key features of the selected studies which used traditional machine learning and deep learning techniques, respectively. This split captures two clear trends identified in the review: ML and DL approaches. Besides being different by themselves, the choice of raw data and features is usually shaped with these trends in mind.

3.1 Study population

All studies here discussed monitored infant movement in some way. It is the data collected via this monitoring that is subsequently analyzed. Papers that did not share common datasets (n=26) summed 3586 infants. These were often preterm or had other comorbidities with a high-risk of developing CDDs. In all studies, infants' GMs were evaluated either by Precthl's (Einspieler & Prechtl, 2005) or Hadders-Algra's (Hadders-Algra, 2004) GMA methodology. Studies analyzing fidgety movement (n=26) used different methodologies, while those analyzing writhing GMs employed only Precthl's method. Optionally, some studies (n=13) assessed infants for neurodevelopmental outcomes after 2 years. More information on study population for every included study can be found in Table 36 and Table 37 in the APPENDIX.

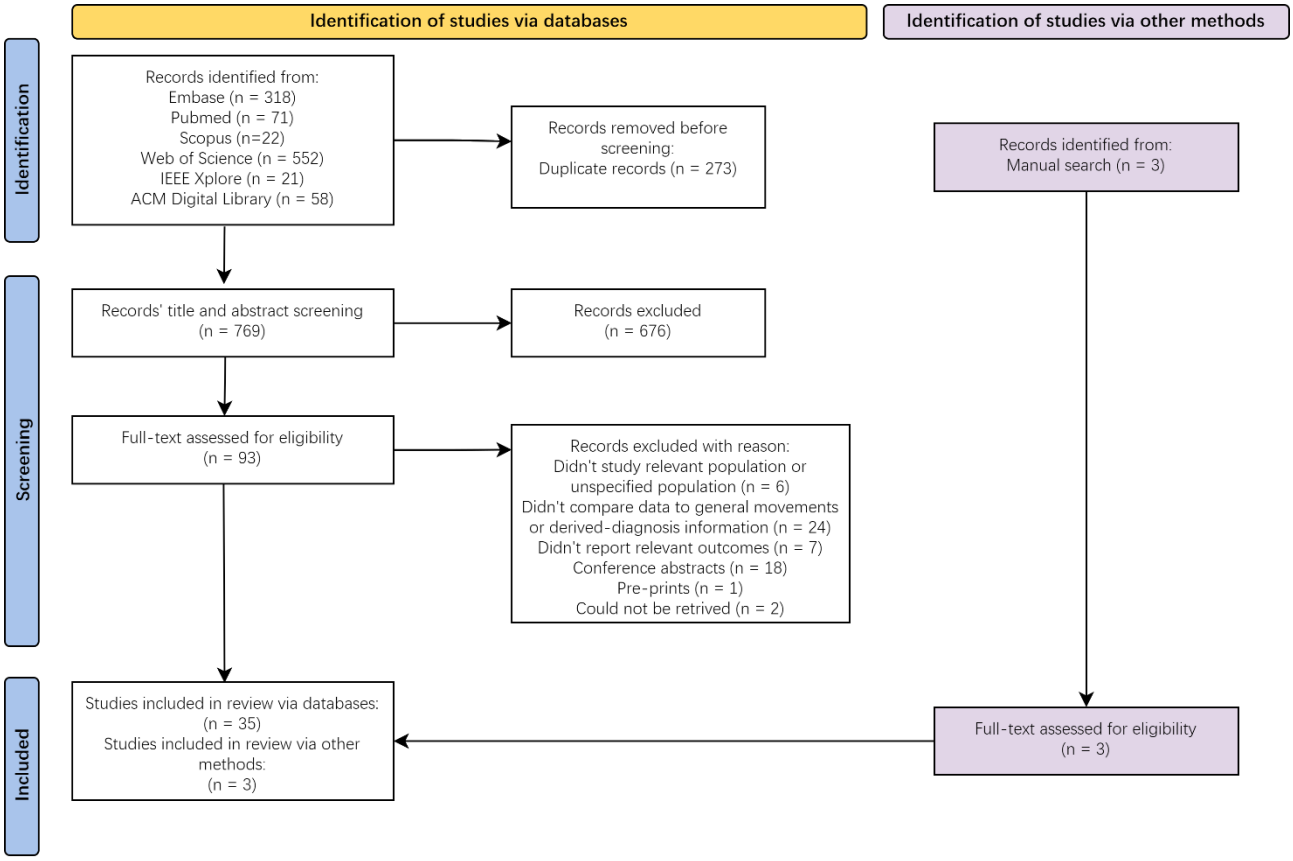


Figure 12 - Identification, screening, and inclusion steps in this review according to the PRISMA flow diagram.

3.1.1 Measurement tools for monitoring movement and raw data

Regarding measurement tools, we divided studies into those using wearables (n=6) and non-wearables (n=32) sensors. Wearables included tri-axial accelerometers (n=3), magnet tracking systems (n=2), and IMSs (n=1). All non-wearable studies used video recordings of the infants' movement. They differ, however, regarding processing. Specifically, studies applied either optical flow (n=9) or pose estimation (n=23) algorithms. Details on all monitoring tools are summarized in the "Raw data" column of Table 6 and Table 7.

3.1.2 Derived features

In most cases, data for subsequent usage was not the raw sensor data, but specific features thought to be relevant. These included motion features – i.e., cross-correlation between opposite-sided joints – and frequency features – i.e., mean velocity, acceleration, jerk for each limb – calculated over temporal windows. Recent studies tend to use raw video frames or

derived pose coordinates in combination with neural networks. Details on derived features are found in the column "Derived features" of Table 6 and Table 7.

Table 6 - Information on ML classification studies' data, methodology, and primary outcome.
Studies with shared raw data, features, or methods are grouped in different colors.

Study	Raw data	Derived features	Method	Primary outcome (%)
Gravem et al., 2012	Triaxial acceleration from 5 accelerometers (t=1hr, s.r.≈19Hz)	166 statistical and temporal features over 1, 2, 4s windows	SVM, DT, Dynamic Bayes + RF **Val.:** 10-fold crossval.	Sn./sp.: **SVM**: 6.9/96.4 **DT**: 10.3/93.9 **DB**: 49.8/76.4
Stahl et al., 2012	RGB video (t=n/s) w/ Huber-L variation optical flow	Absolute motion distance (D), relative frequency (F), magnitude of wavelet coefficients (M)	SVM **Val.:** 10-fold crossval.	SVM sn./sp./acc. for each feature: **D**: 76.7/95.1/91.7 **F**: 85.3/95.5/93.7 **M**: 56/90.7/84.4
Adde et al., 2013	RGB video (t=50s-5min) + GMT	Quantity of motion mean (Q_M) and std. (Q_{SD}), centroid of motion std. (C_{SD}), and their combination (CPP)	LR **Val.:** LOOCV	Sn./sp. for CP and FM class: **Q_M**: 67/58 – 67/65 **Q_{SD}**: 78/51 – 56/56 **C_{SD}**: 100/74 – 89/77 **CPP**: 89/74 – 89/79
Støen et al., 2017		Q, C, C_{SD}	Generalized Linear Mixed Model	**C_{SD}** was lower in normal infants (p < 0.001)
Rahmati et al., 2015	Coordinates from 6 electromagnetic sensors (t=n/s, s.r.=n/s) + RGB video (t=n/s)	Mean and std. of power spectrum as obtained by FFT applied on motion data	PLSR matrix rank **Val.:** LOOCV	Sn./sp./acc. of sensor and video data: **Sensor**: 85/92/91 **Video**: 92/87/88
Orlandi et al., 2018	RGB video (t=>3min) + LDOF	Q_M, mean silhouette orientation, mean velocity minimum (V_{MIN}), mean velocity of silhouette (Vs_M), ratio of velocity on x and y, median velocity of centroid	LR, AdaBoost, LogitBoost, RF **Val.:** LOOCV	Sn./sp./acc./AUC for CP and FM class: **LR**: 44/95/88.19/77 – 41/91/79.53/79 **Ada**: 13/96/85.83/73 – 55/95/85.83/82 **Log**: 25/94/85.04/77 – 48/91/81.10/82 **RF**: 44/99/92.13/82 – 31/94/79.53/83
Raghuram et al., 2019		V_{MIN}, Vs_M, mean vertical velocity (V_Y)	LR	Sn./sp./acc.: 79/63/66
Raghuram et al., 2022	RGB video (t=n/s) + LDOF	Median of Q, Q_{SD}, V_Y, minimum of Q		Sn./sp.: 55.17/79.64
Gao et al., 2019	Triaxial acceleration from 4 accelerometers (t=10min, s.r.=100Hz)	Low dimensional PCA features (d=100)	Discriminative Pattern Discovery (DPD) **Val.:** LOOCV	Sn./sp./acc./pr.: **DPD**: 70/87/80/57 **No-DPD**: 88/68/70/43

Study	Raw data	Derived features	Method	Primary outcome (%)
Ihlen et al., 2019	RGB video (t≈5min) + LDOF	990 statistical and temporal features, pixel center of 6 body parts decomposed via MEMD and Hilbert Huang transformation	CIMA model (PLSR + LDA) **Val.:** Double layer crossval.	Sn./sp./AUC: 92.7/81.6/87
K. D. McCay et al., 2019	MINI-RGBD dataset + OpenPose pose estimation	HOG-based Histograms of Joint Orientation 2D (HOJO2D) and of Joint Displacement (HOJD2D) for *n*-sized windows	kNN (k=1, k=3), LDA, Ensemble **Val.:** LOOCV	Avg. acc. over all tested *n*-sizes, joints, and bin-size: **kNN(k=1)**: 58.33 **kNN(k=3)**: 54.16 **LDA**: 61.84 **Ensemble**: 68.42
K. D. McCay et al., 2021			Ensemble **Val.:** LOOCV	Avg. sn./sn./acc.: 100/100/100
K. D. McCay et al., 2022	MINI-RGBD and RVI-38 dataset + OpenPose pose estimation	HOJO2D, HOJD2D, Histogram of Angular Displacement, Relative Joint Orientation, Relative Joint Angular Displacement, FFT of Joint Displacement Orientation	LR, SVM, LDA, DT, Ensemble, kNN (k=1, k=3) **Val.:** LOOCV	Avg. sn./sp./acc. for all features with their best classifiers for each dataset: **MIN:** 87.50/91.25/90 **RVI:** 75/95.63/92.37
Tsuji et al., 2020	RGB video (t≈442s) + own algorithm optical flow	25 features derived from movement magnitude, balance, rhythm, and movement of body centre	Log-linearized Gaussian mixture network (LLGMN) **Val.:** LOOCV	Acc. for classifying 4 (WM/FM/CS/PR) and 2 classes (Ab./N.): **4:** 83.1 **2:** 90.2
Doroniewicz et al., 2020	RGB video (t≈10min) + OpenPose pose estimation	Factor of movement area and shape, center of movement's area	SVM, LDA, RF **Val.:** LOOCV	Sn./sp./acc.: **SVM:** 71/83/80 **LDA:** 40/94/80 **RF:** 44/93/81
Fontana et al., 2021	Triaxial acceleration from 4 accelerometers (t=10min, s.r.=150Hz)	Cross-correlation of jerk for upper/lower limbs, kurtosis of the acceleration's first PCA component's probability distribution	LR	Sn./sp./AUC: 88/86/89
Wu, Xu, Wei, Kuang, et al., 2021	MINI-RGBD dataset + RGB videos (t≈5min) + PifPaf pose estimation	Angle between joints	Grassberger-Procaccia + Spearman correlation coefficient matrix	Sn./sp./acc.: 100/87.8/91.5
Wu, Xu, Wei, Chen, et al., 2021	MINI-RGBD dataset + PifPaf pose estimation			Sn./sp./acc.: 100/87.5/91.67
Q. Wu et al., 2023	MINI-RGBD and RVI-38 dataset + own method pose estimation	Histogram-encoded joint coordinates and velocities	Affinity Propagation Clustering Model	Sn./sp./acc. on each dataset: **MIN:** 100/87.5/91.67 **RVI:** 100/87.5/89.47

Study	Raw data	Derived features	Method	Primary outcome (%)
Ji et al., 2023	RGB video (t≈630s) + OpenPose pose estimation	72 features derived from wrist and angle velocity, acceleration and angular velocity/acceleration	SVM, DT, RF, GBDT, kNN, AdaBoost, GNB, MLP **Val.:** LOOCV	Avg. AUC over classifiers: 0.851;

s.r.: sample rate; SVM: support vector machine; DT: decision tree; RF: random forest; GMT: General Movements Toolbox; LR: logistic regression; LOOCV: leave-one-out cross-validation; FFT: fast-Fourier transform; PLSR: partial least squares regression; LDOF: large displacement optical flow; PCA: principal component analysis; LDA: linear discriminant analysis; HOG: histogram of gradient; kNN: k-nearest neighbors; GBDT: gradient boosting decision tree; GNB: gaussian naïve Bayes; MLP: multi-layer perceptron.

3.1.3 Classification methods

Data were primarily used for designing and validating classifiers (n=34), or simply for analyzing relationships between variables (n=4). Among classification studies, traditional machine learning methods were employed in 20 papers, the most frequently implemented algorithms being SVMs (n=7), LRs (n=5), and RF (n=5). Fourteen papers used deep learning networks, including convolutional (n=5), graph-based (n=4), and attention-based architectures (n=4). All classifications were binary, and most systems aimed to predict abnormal GM, either in the entire sequences or in video segments. Column “Method” of Table 6 and Table 7 show the specific methods for each study that used ML and DL approaches.

3.1.4 Statistical analysis and study outcomes

Sensitivity and specificity, along with accuracy, are the most commonly used metrics for evaluating a classifier’s performance. The AUC-ROC was also reported in several studies. Comparison between ML and DL approaches’ performance is difficult since both data and processing are vastly distinct (see column “Primary outcome” of Table 6 and Table 7). An emerging exception is the use of the MINI-RGBD and the RVI-38 datasets, which provide frames + 2D/3D coordinates and OpenPose-generated coordinates, respectively. Eleven studies evaluated their classifiers on the MINI-RGBD dataset and 5 on the RVI-38 (6 on both). When first proposed as an annotated infant movement dataset, classifier performance on the MINI-RGBD had already reached 100% accuracy using an Ensemble classifier with 16-binned histograms of the right leg’s displacement (K. D. McCay et al., 2019). An Ensemble classifier proposed by McCay, K. D. et al. (2021) achieved 100% accuracy, sensitivity and specificity values. More recently, Wu, Q. et al. (2023) achieved 100% sensitivity (Sp.: 87.5, Acc.: 91.67) using an Affinity Propagation Clustering model and Zhang, Ho, et al., (2022) got 100% accuracy (Sn.: 100, Sp.:

100) using a graph convolutional network with an attention module. Performance on the RVI-38 traded sensitivity and specificity values, while slightly improving accuracy, which went from 92.37% (K. D. McCay et al., 2022) to 97.37 (H. Zhang, Ho, et al., 2022; H. Zhang, Shum, et al., 2022) – corresponding to an improvement of 2 successfully classified infants. Currently, alongside the 97.37% accuracy level, sensitivity and specificity values are, respectively, 83.33% and 100%. Both Wu. Q et al. (2023) and Sakkos, D. et al. (2021) achieved 100% sensitivity values, although Sakkos' study was evaluated on a smaller (n = 25), previous group from the RVI-38 dataset.

Table 7 - Information on DL and ML/DL both (*) classification studies' data, methodology, and primary outcome. *Studies with shared raw data, features, or method are grouped in different color.*

Study	Raw data	Derived features	Method	Primary outcome (%)
K. D. McCay et al., 2020	MINI-RGBD dataset + OpenPose pose estimation; MINI-RGBD dataset provided 3D coordinates	HOG-based Histograms of Joint Orientation (HOJO2D) and of Joint Displacement (HOJD2D) for n-sized frame windows.	FCNN, Conv1D, Conv2D **Val.:** LOOCV	Avg. acc. for HOJO2D+HOJD2D: **FCNN:** 84.72 **Conv1D:** 81.25 **Conv2D:** 81.25
Zhu et al., 2021		2D-coordinate time-series of body joints	Squeeze-Excitation + attention + Conv2D + FCNN **Val.:** LOOCV	Avg. acc. with and w/o attention: **W:** 91.67 **W/O:** 91.67
Garello et al., 2021**(*)**	RGB video (t≈5min) + DeepLabCut pose estimation	Velocity's cross-correlation of left/right limbs; Skewness of velocity distribution; periodicity of limbs' trajectory; area out of limb's trajectory std. ... + 125 general kinematic/ frequency features	RF, SVM (Polynomial/ Gaussian), FCNN **Val.:** LOOCV	Overall acc. with all/best parameters: **RF:** 56.4/69.1 **FCNN:** 54.5/74.5 **SVM-P:** 50.9/72.7 **SVM-G:** 50.9/78.2
Moro et al., 2022**(*)**	RGB video (t≈8min) + DeepLabCut pose estimation		RF, SVM (Polynomial/ Gaussian), FCNN, LSTM **Val.:** 5-fold crossval.	Avg. acc. for each model: **RF:** 62.6 **SVM-P:** 59.3 **SVM-G:** 59.4 **FCNN:** 58.0 **LSTM:** 59.5
Nguyen-Thai et al., 2021	RGB video (t≈2.5min) + OpenPose pose estimation	2D-coordinate time-series, velocity, acceleration, and travel distance of body joints.	Spatiotemporal attention-based model (STAM) **Val.:** Voting based prediction.	AUC: 0.8187±0.0377
Reich et al., 2021	RGB video snippets (t≈5s) + OpenPose pose estimation	2D-coordinate time-series of body joints	Shallow Multilayer Neural Network **Val.:** 5-fold crossval.	Avg. sn./sp./acc. of best architecture: 88/88/88

Study	Raw data	Derived features	Method	Primary outcome (%)
D. Sakkos et al., 2021	MINI-RGBD and RVI-25 dataset + OpenPose pose estimation	2D-coordinate time-series of body joints	LSTM, Conv1D **Val.:** LOOCV	Sn./sp./acc. for each dataset: **MINI:** 100/87.5/91.67 **RVI:** 100/87.9/91.5
Tong et al., 2022	RGB videos (t=30s) + HRNet pose estimation	2D-coordinate time-series of body joints	Spatiotemporal, Actional-Structural, and Channel-wise Topology Refinement GCNs	Sn./acc. for each model: **ST-GCN:** 77.3/79.0 **AS-GCN:** 88.9/87.0 **CTR-GCN:** 94.7/87.0
Gong et al., 2022	Pmi-GMA dataset + HRNet pose estimation			Acc. for each model: **ST-GCN:** 82.48 **AS-GCN:** 79.84 **CTR-GCN:** 95.54
Groos, Adde, Stoen, et al., 2022	RGB video (t≈5min) + EfficientPose pose estimation	Position, velocity, and distance from neighboring joint of 2D-coordinate time-series of body joints	Ensemble GCN + attention + FCNN **Val.:** 7-fold crossval.	Sn./sp./acc.: 71.4/94.1/90.6
Hashimoto et al., 2022	RGB video (60-210s) + OpenPose pose estimation	Single frames and multi-frame optical flow output (see Farnebäck, 2003 for method)	Conv + FCNN **Val.:** 5-fold crossval.	Sn./pr./acc.: 73.7/78.0/75.2
Luo et al., 2022	RGB video (t=24-653) + OpenPose pose estimation	2D-coordinate time-series of body joints	GCN + Conv2D + Conv1D + LSTM **Val.:** 5-fold crossval.	Acc./F1/AUC: 93.8/94.4/96.9
Zhang, Shum, et al., 2022	MINI-RGBD and RVI-38 dataset + OpenPose pose estimation	FFT + frequency-binning of 2D-coordinate time-series of body joints	GCN + attention + FCNN **Val.:** LOOCV	Sn./sp./acc. for each dataset: **MINI:** 100/100/100 **RVI:** 83.33/100/97.37
Zhang, Ho, et al., 2022				

HOG: histogram of gradient; LOOCV: FCNN: fully-connected neural network; Conv(x)D: x(D)-convolutional neural network; leave-one-out cross-validation; RF: random forest; SVM: support vector machine; LSTM: long short-term memory neural network; GCN: graph-convolutional neural network; FFT: fast-Fourier transform.

4 METHODOLOGY

In this section, we present the methodology employed in this study. This includes a description of our data, the pipeline for preprocessing data, and the adaptation of a graph-based convolutional neural network for identifying abnormal GMs in movement sequences – as well as the experimental setups used for its training, validation, and testing (see Figure 13).

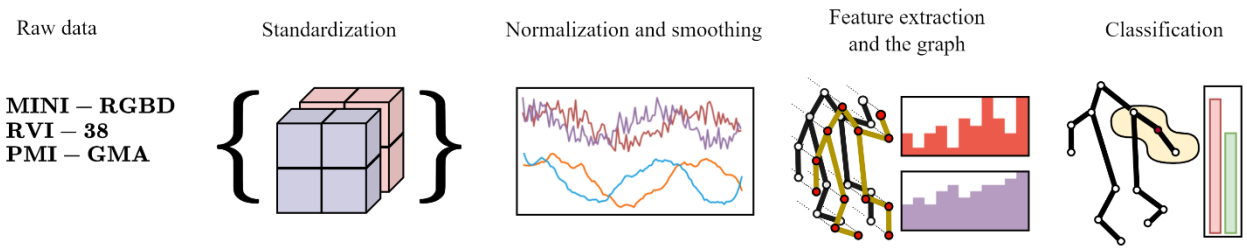


Figure 13 - Methodology overview.

4.1 Data

We collected infant movement data from publicly available datasets containing GMA-annotated sequences of infant movement. Three datasets were selected: the MINI-RGBD (Hesse et al., 2019)[14], RVI-38 (K. D. McCay et al., 2022), and PMI-GMA (Gong et al., 2022; Tong et al., 2022). The **MINI-RGBD** data consists of 12 sequences of synthetically created moving infants. These sequences were first captured from infants up to 7 months of age in the Ludwig Maximilian University of Munich and later registered to the Skinned Multi-Infant Linear body model (SMIL), which renders the synthetic, anonymized, sequences (Hesse et al., 2018). Each sequence is composed of 1000 RGBD frames, and labeled either “normal” for containing movement from typically developing infants or “abnormal” if concerning movements were present, by an independent GMA expert (K. D. McCay et al., 2019). The **RVI-38** dataset comprises 38 videos of 38 different infants aged between 3 and 5 months postterm, collected from the Royal Victoria Infirmary in Newcastle upon Tyne. Each video varies from 40 seconds to 5 minutes of duration, with an average of 3min36s. Videos are labeled “FM+” or “FM-” whether fidgety movements are normal or abnormal, respectively. Lastly, the **PMI-GMA** contains 1120

[14] Dataset available at http.fhg.de/mini-rgbd.

segments of 300 frames each, collected from 87 newborns. For each dataset, the available data consists of pose coordinates outputted by pose estimation algorithms. Sequences from the MINI-RGBD and RVI-38 are available as the raw output of OpenPose (Z. Cao et al., 2021): 2D pose coordinates and corresponding confidence scores. PMI-GMA's sequences were available as the output of the HRNet model (Sun et al., 2019): 2D pose coordinates without confidence scores. Further details are summarized in Table 8.

Table 8 - Summary description of the three datasets used in this study

Characteristic	**Dataset**		
	MINI-RGBD	*RVI-38*	*Pmi-GMA*
Nº of subjects	12 infants (8 normal, 4 abnormal)	38 infants (32 absent, 6 present FM)	87 newborns (64 absent, 23 present PR)
Age interval	< 7 months (not specified unit)	36 - 60 weeks (not specified unit)	Not specified (by newborn definition: < 28 days ChA).
Annotation type	Normal/abnormal **general movement patterns**, as assessed by a GMA assessor in K. D. McCay et al., 2019; Wu, Xu, Wei, Kuang, et al., 2021)	Presence/absence of **fidgety movements (FM)**, as assessed by two GMA assessors	Presence/absence of **poor-repertoire (PR)** movement patterns
Available as	12 2D pose coordinates + confidence scores for 25 joints + label	38 2D pose coordinates + confidence scores for 25 joints + label	2D pose coordinates w/o confidence scores for 17 joints + label

In total, sequences from 137 infants were used, from which 33 contained abnormal GMs and 104 had healthy GMs. For each sequence, GM quality was indicated by a binary label, where $+1$ (positive class) corresponded to abnormal GM sequences. The number of sequences summed 1170 and had variable lengths.

4.2 Pre-processing

Regardless of the dataset, samples (sequences) contained positional data and a label. As different datasets had distinct ways of organizing data, the first step in our pipeline is to standardize their structure – detailed in Section 4.2.1. Then, further processing is applied to each dataset to reduce unnecessary variance of positional data both internally and among each other. Internally, noisy aspects of the data derived from either video capture or pose estimation algorithms, such as missing values and outliers, were addressed and treated by normalization and smoothing techniques. The last row of Table 8 shows how the data is available in each

dataset. Among different datasets, the position and scale of infants were normalized by rescaling, pivoting, and rotation operations. Overall, this allows us to better train our model on mixed data coming from different origins, as particularities of each dataset (e.g., video capture ratio and distance, FPS) and of irrelevant infant features (e.g., body size and orientation) are reduced. These steps are described in Section 4.2.2.

All preprocessing was done in Python v3.9.17 and its following packages: Pandas v1.5.3 and NumPy v1.25 for data handling; Matplotlib v3.7.1 for visualizations; and SciPy v.1.10.1 and Scikit-learn v.1.2.2 for normalization and statistics.

4.2.1 Inter datasets standardization

Data were transformed to represent 2D movement signals for different body joints over time. For each sample, we define a 3D tensor $T_{\mathbf{F,J,C}}$ where $\mathbf{F} = \{\mathbf{f}_0, \mathbf{f}_1, \dots \mathbf{f}_n\}$ denotes the set of indices representing frames, $\mathbf{J} = \{\mathbf{j}_0, \mathbf{j}_1, \dots \mathbf{j}_m\}$ represents the set of indices of joints, and $\mathbf{C} = \{\mathbf{x}, \mathbf{y}\}$ represent the two axes of positional data. Here, n and m denote the total number of frames and joints, respectively. Additionally, samples contain a label: $+1$ for abnormal movement sequence, 0 otherwise. Thus, all 1170 samples, can be represented as a set of tensors $T^{(s)}_{\mathbf{F,J,C}}$ where (s) is the sample ID, specifying the origin dataset and an infant identification number. Figure 14 shows a visual representation of such a tensor.

Frame and joint quantity, n and m, were variable among different datasets. MINI-RGBD samples had a fixed length of $n = 1000$, RVI-38 samples had a variable length, and PMI-GMA had $n = 300$. For now, we keep the original frame quantities. Regarding joints, these datasets had respectively $m = 25$, $m = 25$, and $m = 17$, which corresponds to joints from the torso, limbs, face, and feet. As feet and facial joints do not contribute to GMs – identifiable by limbs' movement –, they were excluded, except the nose joint which loosely tracks head movement. Then, joints shared by all datasets were selected and used as a standard, resulting in 13 joints for each sample (see Figure 15).

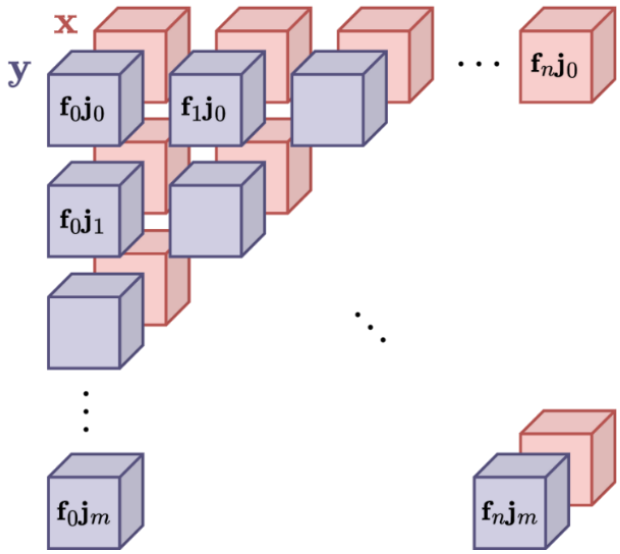
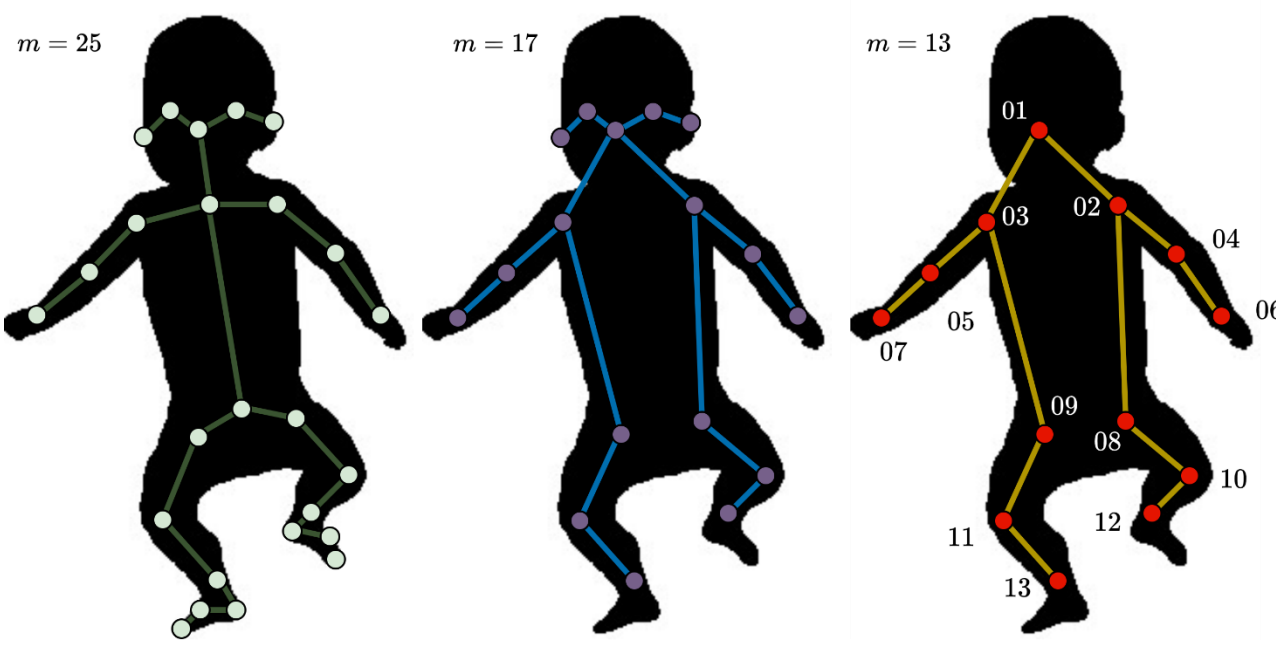


Figure 14 - Tensor structure for a single sample, illustrating three dimensions corresponding to frame quantity, number of joints, and x-y coordinates.



MINI-RGBD dataset PMI-dataset Our work

Figure 15 - Different numbers of joints and their natural connections across datasets.

*Joints from (**left**) MINI-RGBD and RVI-38; (**center**) PMI-GMA, and (**right**) joints used throughout this work overlayed on an infant silhouette mask from the MINI-RGBD dataset.*

Thus, we set the number of joints to 13 ($m = 13$) for all samples, which allows for a more straightforward comparison and interpretation of the subsequent results across all sequences and ensures that each sequence captures the same underlying movements and patterns. Table 9 shows the selected joints' names and an associated numbering system.

Table 9 - The 13 selected joints' numbers and names.
L- and R- prefixes correspond, respectively, to left and right.

Joint nº	Joint name
01	Nose
02	L-shoulder
03	R-shoulder
04	L-elbow
05	R-elbow
06	L-wrist
07	R-wrist
08	L-hip
09	R-hip
10	L-knee
11	R-knee
12	L-feet
13	R-feet

4.2.2 Inner dataset normalization and smoothing

Normalization and smoothing of data were conducted to address inconsistencies in the movement signals, such as pose estimation errors, self-occlusions, and outliers. Large outliers are mostly a result of self-occlusions, especially among leg joints (e.g., knee occluding hip and feet), and present themselves as zeroed values in the movement signal. Otherwise, joint coordinates with an associated low confidence score are also usually outliers. Thus, the first step in our pipeline is to remove zeroes and low confidence values (when available, on MINI-RGBD and RVI-38) and replace them via an interpolation function. Then, we apply the following set of operations on all movement signals to reduce irrelevant variability:

- Rescaling, so that infant size and camera-related aspects (e.g., ratio, distance from infant) are discarded;
- Pivoting, so that at least one of the infant's joint 2D position is fixed throughout the sequence; and
- Rotation, so that infant orientation and specific positions in the 2D space are discarded.

Finally, we set the range of all signals to a specific numerical range and apply a final smoothing function to filter the remaining outliers. Further details are described below.

We first remove all $x-$ and $y-$coordinates with a value of 0, setting it to NaN (not a number). This step yields big gaps in both axes of movement. When dealing with low confidence scores (CIs), the already big gaps of data led us to experiment with different thresholds for discarding these values. MINI-RGBD's and RVI-38's samples had low average CIs, especially for the lower limbs (see Figure 16), and a large threshold for discarding CIs would lead to a disproportionate drop of data points. With that in mind, for each sample, we compute the mean confidence score $S_{\mathbf{conf}}$ of each joint j for every frame f multiplied by a scalar ε for different thresholds. In this context, lower values of ε allow more low-confidence values to remain untouched, while a $\varepsilon = 1$ means that all coordinates with a confidence score that is lower than the joint's mean will be discarded. Then, all coordinates below $S_{\mathbf{conf}}$ are set to NaN.

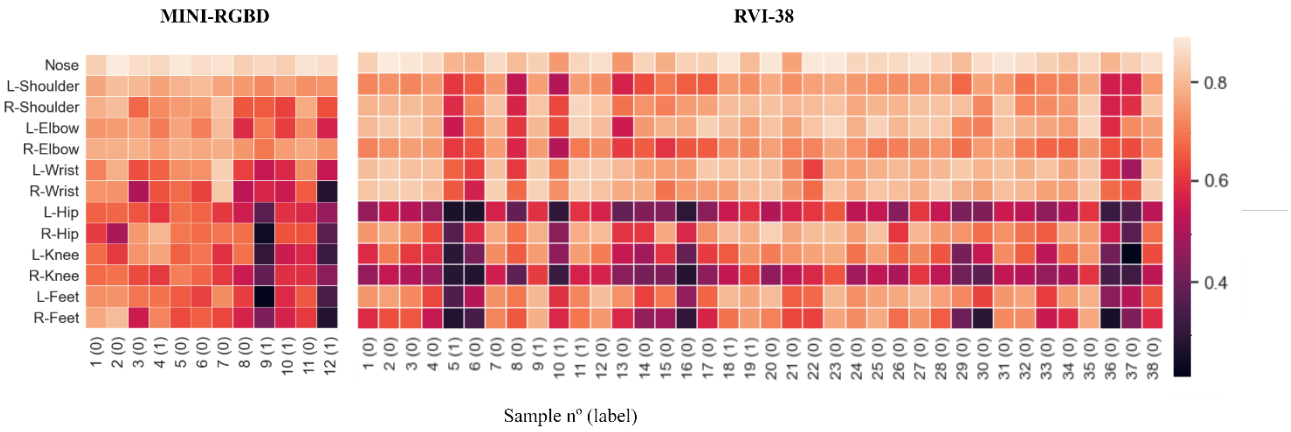


Figure 16 - Mean confidence scores, $S_{\mathbf{conf}}$ of all samples joints' from (**right**) MINI-RGBD and (**left**) RVI-38.

Value in parenthesis on the x-axis corresponds to the label of each sample; 0: negative class, 1: positive class.

A value of $\varepsilon = 0.7$ was found to keep a mean of 25% of the amount of data in each sample and was used throughout the preprocessing. See Figure 17 for an illustration of the drop of low confidence values, and Equation 9 for the computation used, where $c_{(\mathbf{f},\mathbf{j})}$ represents the confidence score of joint $\mathbf{j}$ at frame $\mathbf{f}$.

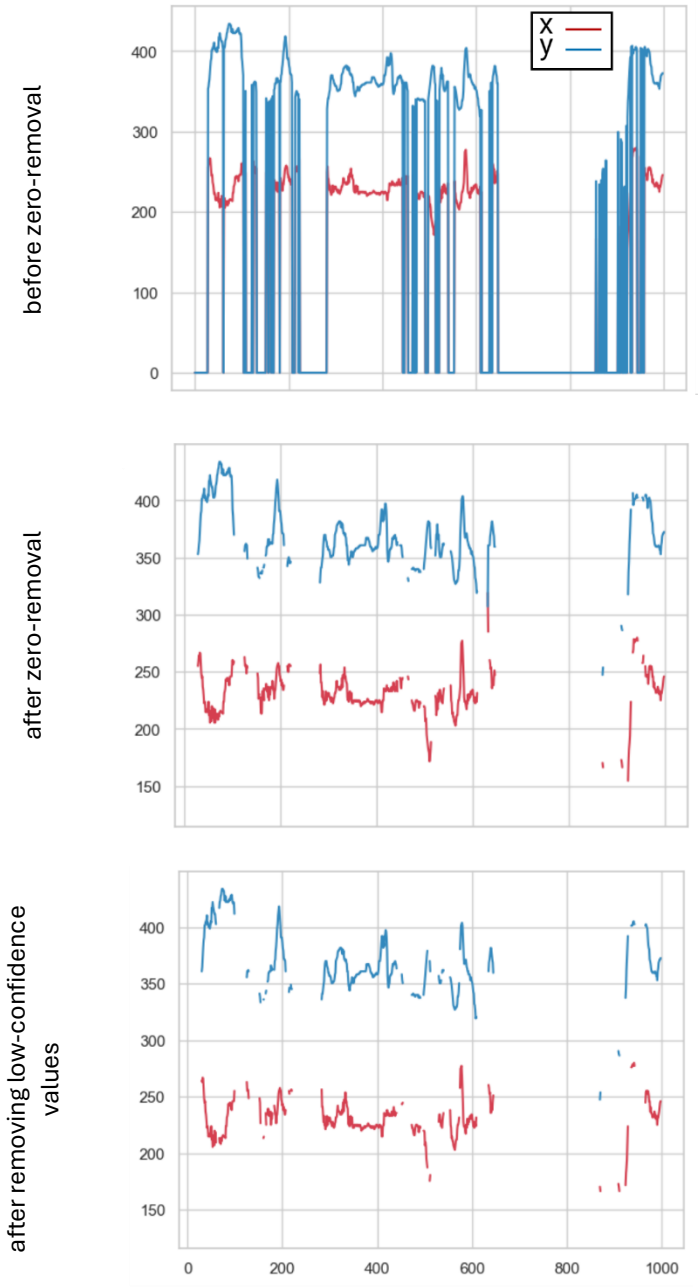


Figure 17 - X — and Y — movement signals from a low-quality MINI-RGBD sample for the left knee joint at different stages of pre-processing.

$$S_{\mathbf{conf}} = \frac{1}{n}\sum_{\mathbf{j}=1}^{n} c_{(\mathbf{f},\mathbf{j})} \times \varepsilon \tag{9}$$

The next step is to fill gaps through interpolation. Different methods were tested, including simple linear, quadratic, and cubic interpolation, Akima interpolation (Akima, 1970), and Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) (Fritsch & Butland, 1984). All

methods except PCHIP and linear interpolation tended to overshoot datapoints while interpolating on large gaps, i.e., connecting lines would go way over/below the datapoint, before connecting. Although linear interpolation can be considered a safe option, commonly used across the literature (Moro et al., 2022; Q. Wu et al., 2023; H. Zhang, Shum, et al., 2022), PCHIP is more well-suited to monotonic signals such as ours and better captures the signal's trends without large oscillations. Thus, we are the first study in the GMA automation literature – to the best of our knowledge – to use the PCHIP method (see McCay et al., 2022 for an application of Akima interpolation). Figure 18 shows the result of some of the interpolation methods experimented with.

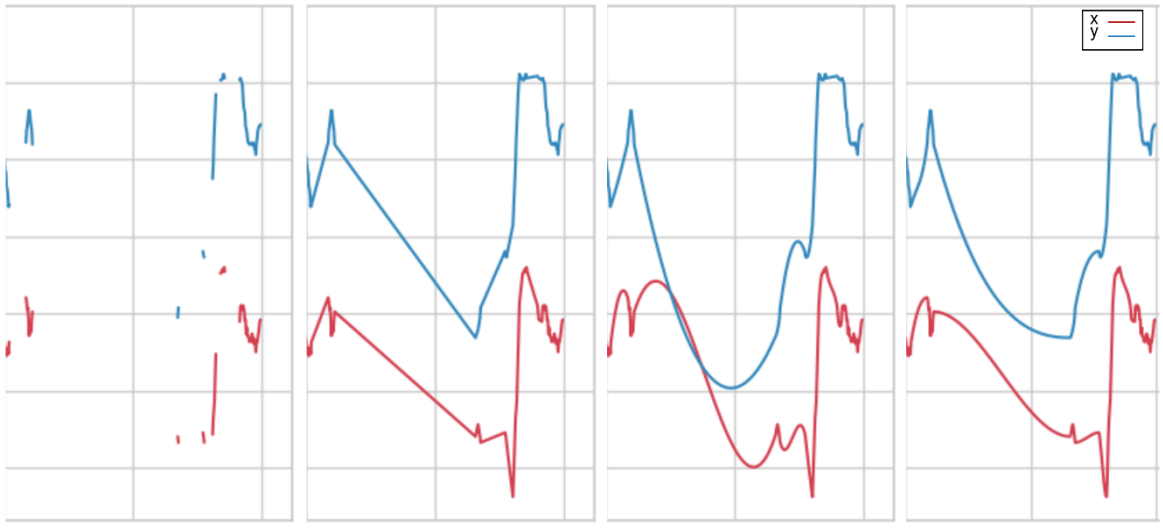


Figure 18 - An illustration of different interpolation methods on a segment of x − and y − movement signals.

Once the signal is interpolated, unwanted variability between samples still remains – such as infant size and orientation. These are irrelevant for the assessment of general movements and the subsequent classification into normal or abnormal sequences, and only contribute as noise for our task. So, to remove such variability, we pass all datasets' signals through a three-phase process: rescaling, pivoting, and rotation. This assures all infant's motion representation in their respective signals are of the same size, positioned in the same place, and oriented in the same direction. Rescaling was done by finding the vector $\vec{v}$, which connects the nose joint with the

midpoint between the left and right hip, and using its length $\|\vec{v}\|$ (frame-invariant) to divide all $xy-$ coordinates. We pivot all coordinates around the nose coordinate by subtracting its value from all other coordinates, for each frame. Then, rotation is achieved by aligning all coordinates with reference to the angle θ between $\vec{v}$ and the horizontal $\mathrm{x}-$axis, resulting in a standard orientation for all samples. A plot of pose coordinates for a single frame in the first step of preprocessing and after the three-phase process can be found in Figure 19.

High-frequency oscillations are consistently found throughout all the movement signals, and represent fast, abrupt changes in position otherwise impossible to occur naturally, such as a leg-length spasm in 200ms. To address these, we smoothed each axes signal by a rolling window median filter. We set the window to 5 frames, and filter out values that exceed the computed median (see Figure 20). By applying this technique, we are able to filter out most of the remaining noise and outliers, despite having no clear confirmation on whether filtered coordinates were a result of natural movement or not.[15]

Recall about the duration and frame quantities of RVI-38's samples, which varies from 40s to 5min, averaging 3min and 36s. To standardize this, we split these sequences into 1000 frames segments and discard segments with less than 1000 frames (for instance, a sequence of 5583 frames would be segmented into 5 segments of 1000 frames, and 583 frames would be discarded). This process "expands" RVI-38 size from 38 samples to 124, while keeping the positive class largely unchanged, from 18% to 16%. Finally, we normalize all signals to a common range of $[0, 1]$ using *min-max* normalization, resulting in 1256 pre-processed samples which will be used for feature extraction.

[15] When inspecting data quality, we created pose data animations by plotting the $xy-$ coordinates and their connective lines along a sequential (frame) axis. While it was possible to identify clear outliers resulting from miss-estimations from the pose estimation algorithm, we were not able to manually inspect all frames of all samples. That said, it is not certain that all datapoints here considered noise/outliers were not a result of natural movement, a limitation scarcely admitted by related studies.

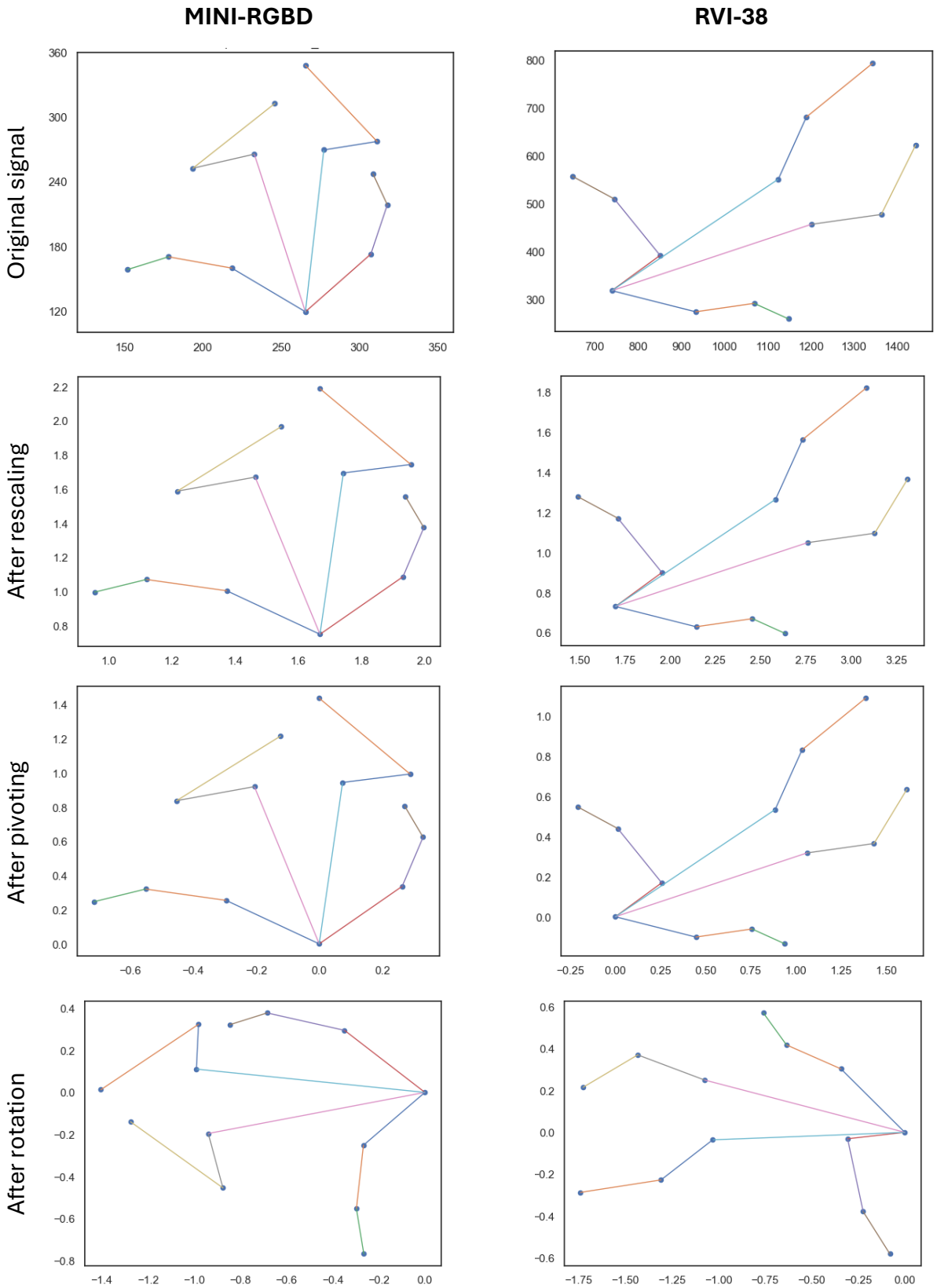


Figure 19 - Random samples for the MINI-RGBD and RVI-38 datasets during the three-phase process of variability removal.

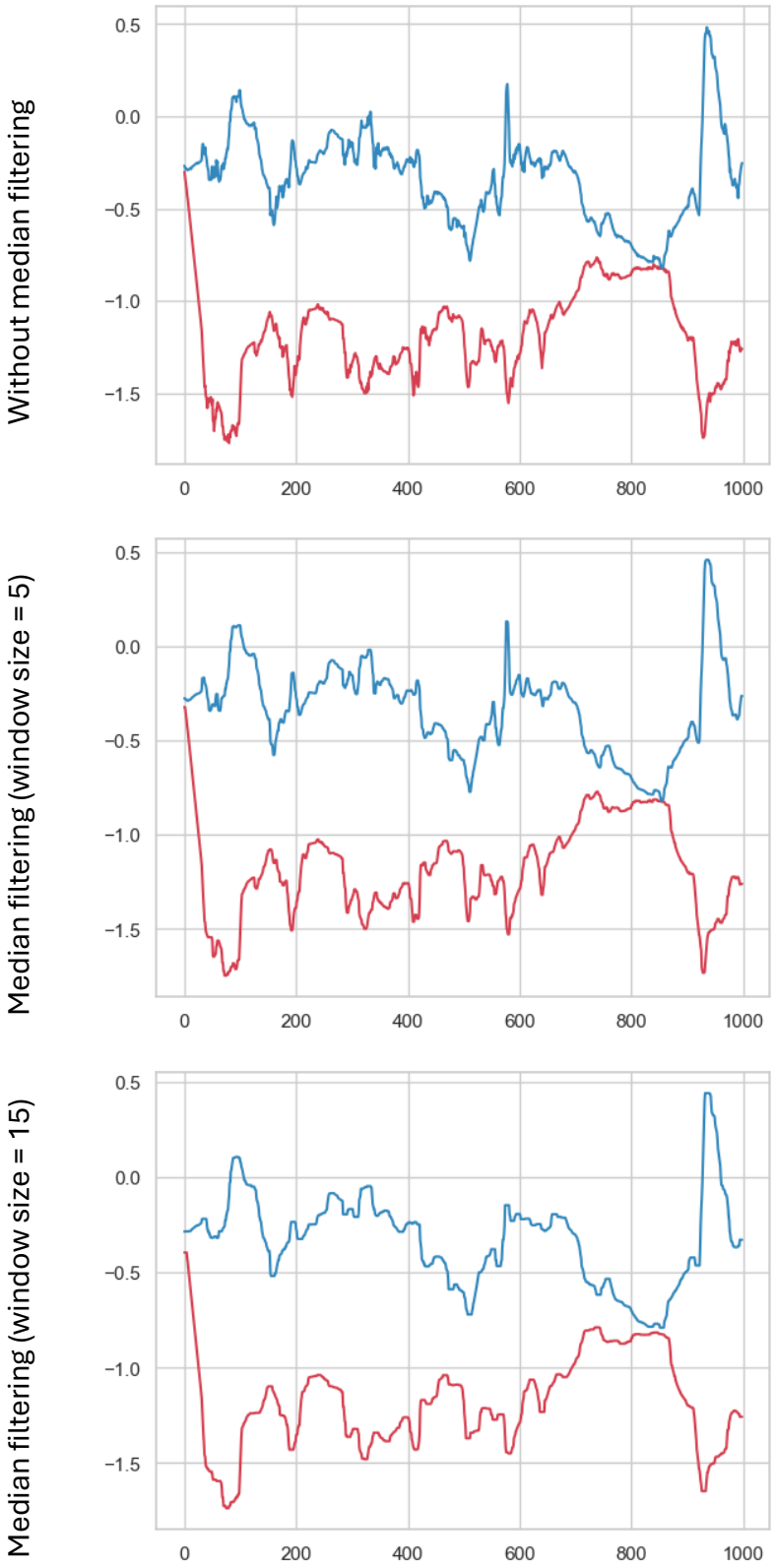


Figure 20 - A random sample's joint movement signal after applying different filtering methods.

*(**Top**) without median filtering and with median filtering with window sizes of (**middle**) 5, as applied, and (**bottom**) 15, for comparison.*

4.3 Feature extraction, histogram encoding, and the graph

We aim to capture both movement magnitude and direction. Since we are dealing with deep neural networks, we believe that high-level features will be represented by the hidden layers of our architecture, and thus its input should consist of low-level features. Magnitude is captured by a 2D displacement vector for each joint between frames, and direction is captured by computing the motion orientation of each joint between frames. Furthermore, we use sliding windows and a stride parameter – the distance between consecutive windows – to set the frame intervals on which feature extraction happens. For a joint **j** in frame **f** at position $P_{(x,y)}$, with window size w and stride size s, the Euclidean distance and motion orientation are computed, respectively, by Equations 10-11, and illustrated on Figure 21.

$$\Delta P_{\mathbf{j}}^{\mathbf{f},w} = \left\|\left(x^{\mathbf{f}}, y^{\mathbf{f}}\right) - \left(x^{\mathbf{f}+w}, y^{\mathbf{f}+w}\right)\right\| \sqrt{\left(P_{(\mathrm{x})_{\mathbf{j}}}^{\mathbf{f}} - P_{(\mathrm{x})_{\mathbf{j}}}^{(\mathbf{f}+w)}\right)^2 + \left(P_{(\mathrm{y})_{\mathbf{j}}}^{\mathbf{f}} - P_{(\mathrm{y})_{\mathbf{j}}}^{(\mathbf{f}+w)}\right)^2} \tag{10}$$

$$\theta_{\mathbf{j}}^{\mathbf{f}} = \tan^{-1}\left(\Delta P_{(y)_{\mathbf{j}}}^{\mathbf{f},w}, \Delta P_{(x)_{\mathbf{j}}}^{\mathbf{f},w}\right) \tag{11}$$

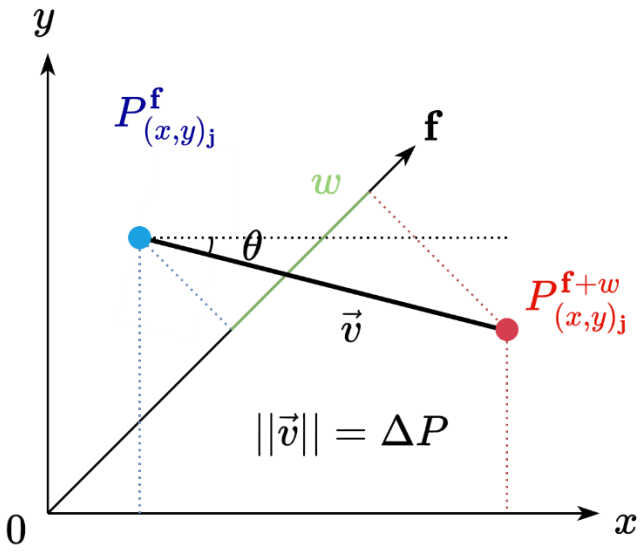


Figure 21 - Illustration of the feature extraction process.

For a joint $\boldsymbol{j}$ *in frame* $\boldsymbol{f}$ *at position* $P^{f}_{(x,y)_j}$ *(in blue) with a window size w,* ΔP *is the Euclidean distance between points at different places in the* $\boldsymbol{f}$*-axis;* θ *is the angle between the starting point of vector* $\vec{v}$ *and the horizontal* $x-axis$.

Computing these features for every frame would not only render a large quantity of features, i.e., a large feature space, but could also encourage the model to overfit – given that these features would correspond to very small changes in displacement and orientation, often unrepresentative of GMs. By applying window-sliding with window and stride set to 30 frames (1s), we are able to reduce the feature space considerably, ignoring the small changes in orientation and magnitude that happen in unobservable ranges of 300ms-500ms while retaining those intervals commonly noted by GMA assessors ($\approx 1-5s$) (Einspieler & Prechtl, 2005; Hadders-Algra, 2004). After this step, each sample is transformed from its previous form $T_{\mathbf{F,J,C}}$ to $T_{\mathbf{W,J,F}spc}$, where $\mathbf{W} = \frac{\mathbf{F}-w}{s} + 1$ and $\mathbf{F}spc = [\Delta P, \theta]$, that is, the number of frames is reduced to the number of resulting windows, joints remain untouched, and 2D coordinates are replaced by the abovementioned features.

To further summarize the feature space, we follow McCay's *et al*. (2021, 2022) method of histogram encoding. We partition each feature vector into b bins delimited by the minimum and maximum value of the vector and count occurrences falling within each bin. By normalizing these counts, a histogram represents the probability distribution of the feature vector, and further reduces feature space from the number of computed windows to the number of bins, b, as seen in Figure 22. These histograms represent the temporal dimension of each sequence, as feature values that vary significantly over time will be reflected in the histogram's shape and distribution.

Finally, we construe a graph $G = (V, E, \mathbf{X}_V)$ where V is the set of 13 joints, E is the set of 12 connecting bones, and the node features – histograms per joint – are represented by $\mathbf{X}_V = \{A \in \mathbb{R}^{13\times10}\}$. The graph is undirected: edges have no direction. The connectivity information of nodes is stored in a adjacency matrix $\mathbf{A} = \{A \in \mathbb{R}^{13\times13}\}$, where connected nodes are denoted by ones, and unconnected by zeroes. In our graph, nodes are connected to themselves via self-loops by adding the identity matrix onto $\mathbf{A}$. If we were to use the raw $\mathbf{A}$ matrix, convolution would generate feature embeddings with large magnitudes for nodes with many neighbors, and small magnitudes for nodes with few neighbors. As layers stack, this may become a numerical stability problem where weights achieve unmanageable magnitudes. To prevent that, we

normalize the adjacency matrix by how many neighbors a single node has, and how many neighbors each neighbor have (see Figure 23) (Kipf & Welling, 2017)[16].

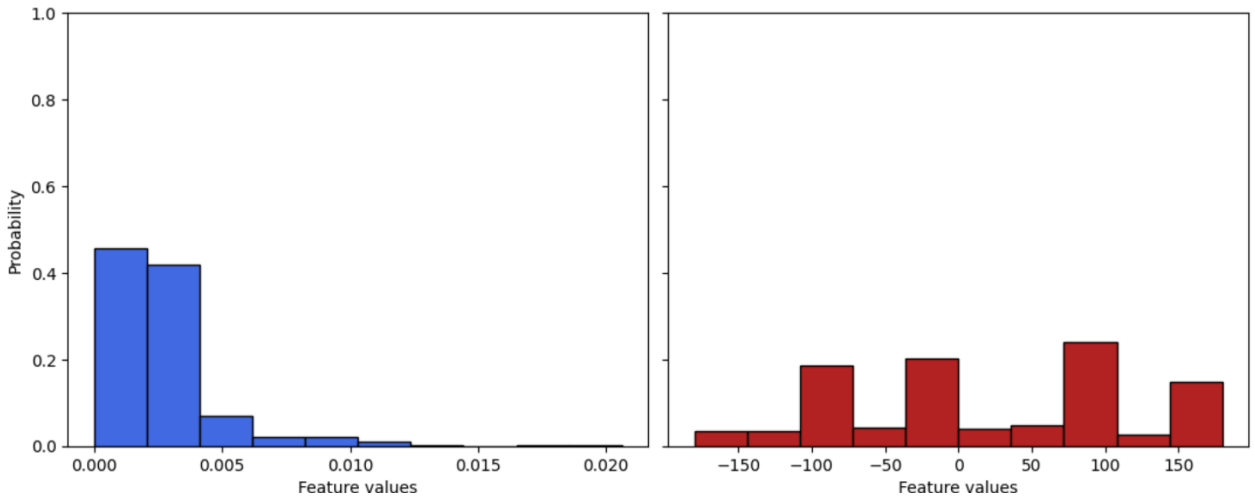


Figure 22 - Normalized histogram-encoded features for one joint of a random sample.

*The y-axis represents the relative frequency of occurrence and the x-axis represents the feature values. Features are (**right**) ΔP and (**left**) θ.*

$$\begin{vmatrix}
1 & .71 & 0 & 0 & .29 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
.71 & 1 & .71 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & .71 & 1 & .71 & 0 & .29 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & .71 & 1 & 0 & 0 & .71 & 0 & 0 & 0 & 0 & 0 & 0 \\
.29 & 0 & 0 & 0 & 1 & .29 & 0 & 0 & .29 & 0 & 0 & 0 & 0 \\
0 & 0 & .29 & 0 & .29 & 1 & .29 & 0 & 0 & .29 & 0 & 0 & 0 \\
0 & 0 & 0 & .71 & 0 & .29 & 1 & .29 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .29 & 1 & .29 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & .29 & 0 & 0 & .29 & 1 & .29 & 0 & .29 & .29 \\
0 & 0 & 0 & 0 & 0 & .29 & 0 & 0 & .29 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .29 & 0 & .29 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.29 & 0 & 0 & 0.29 & 1
\end{vmatrix}$$

Figure 23 - Representation of the normalized adjacency matrix, A.

The pair of features and normalized adjacency matrix ($\mathbf{X}_V$, $\mathbf{A}$) constitute the final representation of a single sample, and will be used as the input for our neural network architecture, described in section 4.4. Figure 24 illustrates a summary of the feature extraction process, from the

[16] A detailed explanation is out of scope. Broadly, this normalization technique leaves unconnected edges as zeroes, but transforms connected ones to be the inverse of the square root of multiplying the number of direct neighbors and indirect (neighbors of neighbors). This allows peripherical nodes (e.g. feet and hands) to have similar weights to central nodes (e.g. shoulders). See (Kipf & Welling, 2017).

extraction of motion signals to the computing of window-wise features, and finally the histogram-encoding of these features into *b* bins.

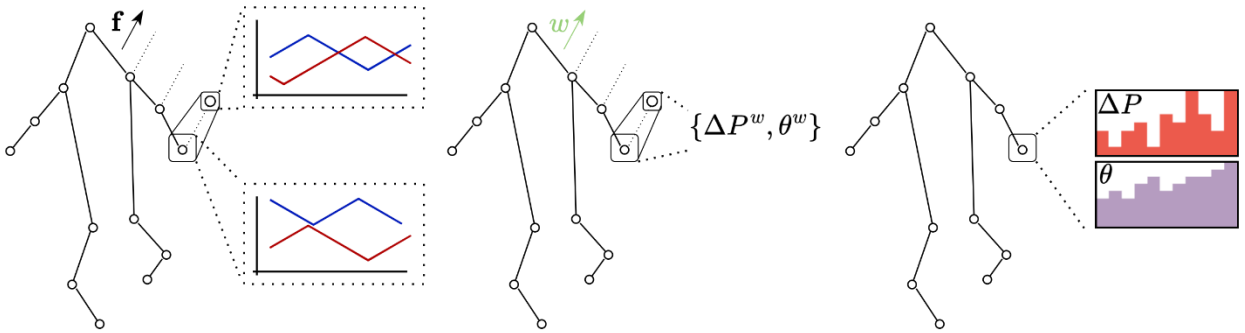


Figure 24 - Overview of the feature extraction procedure for one arbitrary joint.

4.4 Multi-stage spatio-temporal graph convolutional network architecture

Our model architecture is adapted from the Multi-stage spatio-temporal graph convolutional network (MS-STGCN) proposed by (Filtjens et al., 2023), which extends on the architecture of ST-GCNs (Yan et al., 2018). Originally, the MS-STGCN is an action segmentation system whose final output is a mask-like vector containing the model predictions for each frame. This means that its original goal is to identify the start and end point of single actions (e.g. jumping, yawning, etc.), and its output is a vector with size equal to the length of the inputted sequence, where action classes are encoded. For instance, a sequence of 10 frames where jumping occurs in frame 2-4 and yawning in frames 6-7 could be represented by the vector $[0,1,1,1,0,2,2,0,0,0]$, where jumping and yawning are encoded as ones and twos.

Here, we adapt the network so that it predicts a label for each of the histogram-encoded bins, which represent summary aspects of each sequence's temporal dimension. As we still aim for binary classification of sequences containing *either* normal or abnormal movements, a threshold is applied to this prediction vector transforming it into a binary prediction. The inner architecture and implementation details are described below.

4.4.1 Model architecture

An overall picture of the model structure is illustrated in Figure 25. For the following explanation, let *BS* = batch size, *FT* = number of features, *HB* = number of histogram-encoded bins, *J* = number of joints, and $\mathbf{ft_m}$ = number of feature maps. As a first step, the feature vector $\mathbf{X}_V$ of shape (BS, FT, HB, J) passes through a batch normalization (BN) layer, which standardizes feature distribution along the batch. A reshaped vector $(BS, FT \times J, HB)$ is expanded through a 1D convolution layer which outputs a $(BS, \mathbf{ft_m}, HB, J)$ vector.

A series of 10 ST-GCNs units is then applied, which uses the adjacency matrix $\mathbf{A}$ to perform convolution on both spatial (joint-wise) and temporal (bin-wise) dimensions. First, spatial convolution is done by a spatial module ($\mathbf{SpC_m}$), which (i) expands the feature vector according to a kernel and stride size; (ii) use Einstein summation to aggregate information from neighboring nodes, again reducing the feature vector to $(BS, \mathbf{ft_m}, HB, J)$, and (iii) applies batch normalization.

Secondly, temporal convolution happens inside the temporal module ($\mathbf{TpC_m}$), which (i) applies the ReLU activation function element-wise; (ii) does temporal convolution on the histogram-encoded features – which serves as an abstraction on the raw, sequential frames – with a specific kernel and stride size; and (iii) applies batch normalization. Finally, a dropout layer with probability $p = 0.2$ is applied and the ST-GCN output is concatenated with a residual[17] (the feature vector earlier generated by the 1D convolution).

After 10 ST-GCN iterations, we apply pooling to the spatial dimension of the vector, resulting in a $(BS, \mathbf{ft_m}, HB)$ vector. This aggregates the high-level spatial embeddings determined by the previous layer, and functions as a down-sampling technique towards our binary classification.

Our model's last step is to apply convolution on the $\mathbf{ft_m}$ dimension, which reduces the feature vector to $(BS, number\ of\ classes, HB)$, which is then passed through a *soft-max* function.

[17] Residuals, or "skip connections", were first proposed as part of the ResNet architecture (He et al., 2016) for facilitating the optimization process, as well as regularizing an architecture's activations by skipping layers. For a detailed explanation on using residuals and their advantages see (Shafiq & Gu, 2022).

The final output for a single sample, thus, is a mask containing the probabilities of each of the HB-bins pertaining to the negative or positive class. However, as we intend for binary classification of entire sequences (i.e., not singular actions), we set a threshold thr so that a sequence is classified as positive only if the ratio of 1s in the mask exceeds thr.

4.4.2 Implementation details

The threshold value thr was set as a hyperparameter, and evaluated on our optimization experiments. For the spatial module, $\mathbf{SpC_m}$, the kernel size and stride size are set to 2 and 1, respectively. $\mathbf{TpC_m}$'s same parameters are 1 and 1, respectively. The number of feature maps $\mathbf{ft_m}$ and bins were experimented for optimization (see Section 4.5.2). Our loss function is a combination of binary cross entropy (see Equation 4) and mean squared error (MSE), as proposed by (Filtjens et al., 2023).

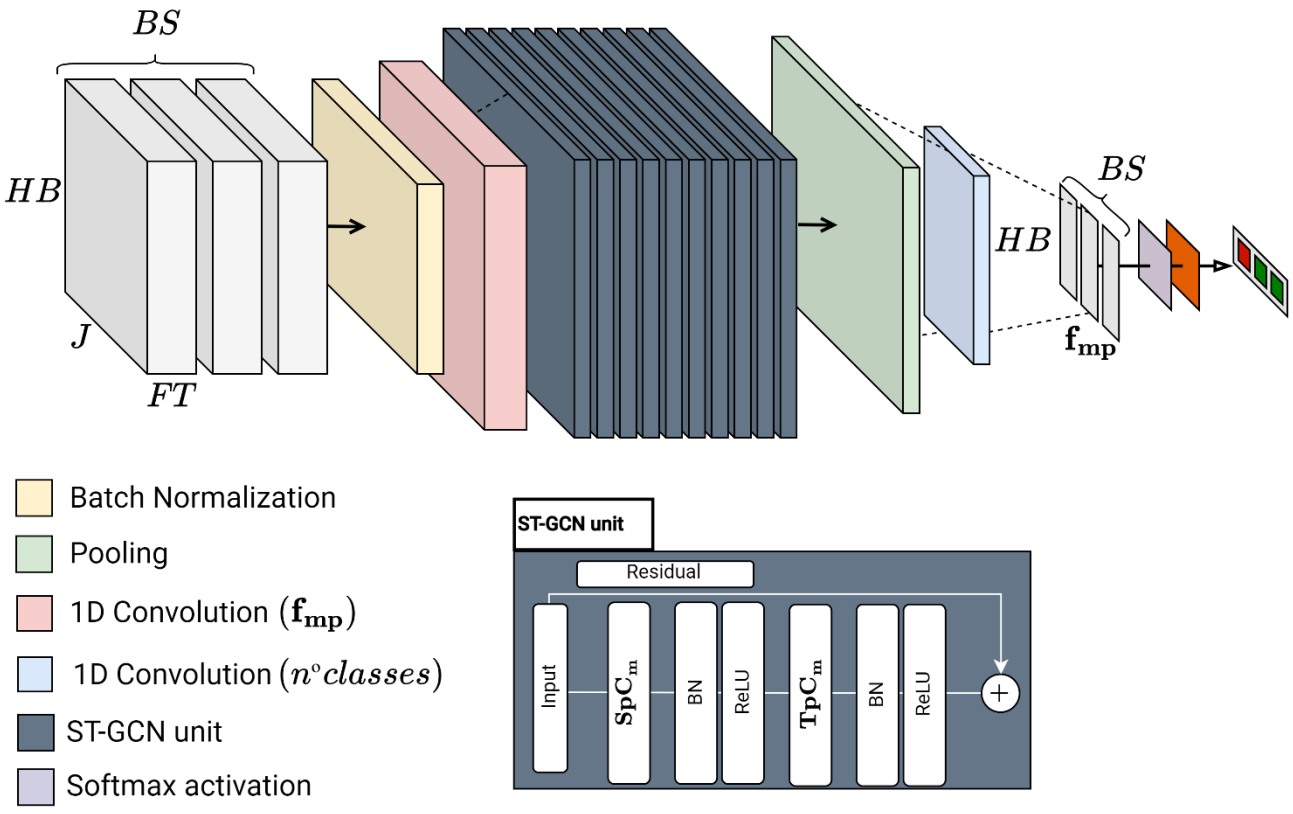


Figure 25 - Model architecture.

We used Adam (Kingma & Ba, 2015) as our optimizer, and experimented with different weight decay parameters, as well as with different learning rate values. Learning rate, or α, controls the rate at which the model learns; weight decay, on the other hand, prevents the model from learning large weights that might lead to overfitting. We set a number of epochs equal to 50 for all experiments. Batch sizes were different for each experimental design, and is reported in the next section.

The model and all related experiments were implemented using the PyTorch v2.2.1 module of Python v3.9.17. All experiments were run on a NVIDIA Tesla T4 GPU, with 16GB of RAM, 2560 CUDA cores, through the Google Colab service.

4.5 Experiments design

This subsection describes our methodology for arranging the available data into different setups, splitting data into training, validation, and testing sets, optimizing a selection of our network's hyperparameters, and reporting relevant metrics.

4.5.1 Experiments setups

We arranged the three available datasets in four different ways, resulting in distinct experimental setups. These setups allow us to compare our results with the related literature, which has the most commonly reported results on single datasets (D. Sakkos et al., 2021; K. D. McCay et al., 2019, 2020, 2021; K. McCay et al., 2022; Q. Wu et al., 2023; Y.-C. Wu et al., 2021; H. Zhang, Ho, et al., 2022; H. Zhang, Shum, et al., 2022), while also exploring all the potential of publicly available data. We describe the training, validation, and testing information of each setup on Table 10. All splits are stratified with regard to class. Those named "Single-[*dataset*]" refer to setups whose data comes from only one dataset. Setups with regular splits had 80% of data separated for training, and the remaining 20% was additionally split 80/20% to form validation and testing data. An adapted Nested-LOOCV was used when data samples were significantly small, as in Single-MINI. It worked by separating one sample for testing, and then further splitting the remaining $N - 1$ samples into training and validation by 80/20% splitting. For Single-MINI, we assured that the validation split would always contain one positive class and one negative class; for the Single-RVI setup, at least 3 positive instances were always present in the validation split.

Table 10 - Experimental setups' information regarding total number of samples, class balance, method used for data splitting, and batch size.

Setup	Total	Training/Validation/Testing	Split method	Batch size
	Nº of samples (prevalence of positive class)	*Nº of samples*		
Single-MINI	12 (0.33)	9 / 2 / 1	Nested-LOOCV	1
Single-RVI	124 (0.16)	99 / 20 / 5	0.8/0.2/0.2 (out of val.) split	13
Single-PMI	1120 (0.50)	896 / 179 / 45	0.8/0.2/0.2 (out of val.) split	16
MINI+RVI+PMI	1256 (0.46)	1004 / 201 / 51	0.8/0.2/0.2 (out of val.) split	64

4.5.2 Hyperparameters optimization

We performed a random search (Bergstra & Bengio, 2012; Liashchynskyi & Liashchynskyi, 2019) 40 times on a pre-defined set of values for **learning rate**, **weight decay**, **video threshold**, **number of feature maps**, **number of bins**, and **pooling operation** (see Table 11). Thus, 40 models with random combinations of hyperparameters are created according to each experiment. Our aim is to find the combination of hyperparameters that yields the best results on the validation set of each setup. We evaluate metrics on each trial of the random search, then select the best performing one for evaluation on the unseen data of the testing split. The optimization procedure is done in each setup, separately, and metrics are reported.

Table 11 - Hyperparameter's names, meanings, and set of values used throughout hyperparameter optimization.

Hyperparameter	Meaning	Values
Learning rate (α)	Rate at which the model learns from data	$\{0.01, 0.005, 0.001\}$
Weight decay (λ)	Amount of penalization to large parameter values	$\{1e-3, 1e-4, 1e-5\}$
Video threshold (*thr.*)	Ratio of positive mask values needed for classifying a sequence as positive	$\{0.3, 0.4, 0.5\}$
Nº of feature maps (ft_m)	Quantity of filters applied to input data	$\{32, 64, 128\}$
Nº of bins (*b*)	Quantity of bins used during histogram-encoding	$\{6, 12, 18\}$
Pooling operation (*pool*)	Type of pooling used in the model	{'max', 'mean'}

4.5.3 Reported metrics

We follow the related literature on reporting accuracy, the AUC-ROC, sensitivity, and specificity. As seen in section 2.2.5, these are standard metrics for evaluating the performance of classification systems. We also investigate the precision (PPV), F1 score and the PR-AUC, as these can be more meaningful when dealing with unbalanced datasets such as the MINI-RGBD and the RVI-38. A summary of these metrics, their meaning, and their formula are listed for reference in Table 12.

Table 12 - Reported metrics and their meaning.
"Unhealthy" subjects, in this context, refer to sequences containing abnormal GMs (i.e., positive classes).

Metric	Meaning	Formula
Accuracy	Total number of instances correctly classified	$TP + TN/TP + TN + FP + FN$
Sensitivity (Recall)	Number of unhealthy instances correctly classified among unhealthy subjects	$TP/TP + FN$
Specificity	Number of healthy instances correctly classified among healthy subjects	$TN/TN + FP$
AUC-ROC	Area under the ROC curve, representing a trade-off between sensitivity and 1-specificity	Numerical integration for approximating area under curve
Precision (PPV)	Number of unhealthy instances correctly classified among all unhealthy predictions	$TP/TP + FP$
F1 Score	Harmonic mean of precision and sensitivity	$2 \times (PRC \times RCL/PRC + RCL)$
PR-AUC	Area under the PR curve, representing trade-off between precision and recall.	Numerical integration for approximating area under curve

TP: true positive; TN: true negative; FP: false positive; FN: false negative; AUC-ROC: area under the receiver operating characteristic curve; PPV: positive predictive value; PRC: precision; RCL: recall; PR-AUC: area under the precision-recall curve.

5 RESULTS AND DISCUSSION

This section presents our experiments on validating the use of the adapted MS-STGCN to predict general movements patterns. Specifically, our model predicts a positive class for sequences that contain abnormal GMs, and negative otherwise. As a result from our pre-processing pipeline, data from different datasets have been processed so that only variability pertinent to general movements' behavior remains, while differences on size, data capture setups, and specific pose algorithms were reduced. Furthermore, by dropping segments of low quality (e.g. low confidence values) and reconstructing the movement signals, we address some of the bad quality issues with the data, and increase the chances that our model is able to generalize and learn properly.

For each experimental setup, we first describe and discuss the performance of our optimal models for each fold/iteration. Secondly, we analyze results achieved during hyperparameter optimization, which justifies the choices made for the optimal models. As datasets are unequal in size and proportion of classes, results varied significantly. Our last experiment, MINI+RVI+PMI, was thought of in order to better represent the model's learning capabilities, and is discussed in further detail.

Performance metrics for all hyperparameters during the optimization procedure for each experimental setup are reported below. For every experimental setup we present summary metrics for each hyperparameter impact throughout *all* the training runs, the distribution of the F1 score for each hyperparameter, and the hyperparameters of the best performing models together with their performance on the testing split.

5.1 Single-MINI

This experiment accounts for the MINI-RGBD dataset, which contains 12 samples, 4 of which are positive. We used an adapted Nested-LOOCV to train 40 models with random selected hyperparameters on each of the 12 folds, select the best performing model (i.e., optimal model) in each fold, retrain a model with the best hyperparameters, and assess it in the remaining one sample of the testing split. The true class of the test sample can be derived from the metrics in Table 13 by looking at whether sensitivity or specificity are unachievable ("---"), corresponding to being a negative or positive true class. We achieve an average accuracy of 75%, with a

balanced sensitivity and specificity of 75% –corresponding to 2 missed predictions for each class.

Overall, our results on the MINI-RGBD are below the most recent state-of-the-art literature, which have yielded performances of 100% with Attention-based neural networks on LOOCV protocols. (H. Zhang, Ho, et al., 2022; H. Zhang, Shum, et al., 2022). By using a Nested-LOOCV, we believe our results better represent the model capabilities, which reached an average of 75% accuracy, sensitivity, and specificity in the test split across all folds (see Table 13). The table also shows the optimal hyperparameter combinations of each model. Since Single-MINI testing split only contains one sample, we report only accuracy, sensitivity, and specificity.

Table 13 - Optimal model for each fold of Single-MINI and their performance on the testing split.

Fold	**Hyperparameter values**						**Performance on testing split (n=1)**		
	α	λ	*thr.*	ft_m	*b*	*pool*	Acc.	Sns.	Spc.
1	0.01	0.01	0.5	64	6	max	1.	---	1.
2	0.001	0.001	0.4	128	6	avg	1.	---	1.
3	0.001	0.0001	0.4	128	12	max	1.	---	1.
4	0.01	0.0001	0.3	64	6	avg	1.	1.	---
5	0.01	0.01	0.3	128	18	avg	1.	---	1.
6	0.0001	0.01	0.3	64	18	avg	0.	---	0.
7	0.0001	0.01	0.4	32	12	avg	1.	---	1.
8	0.01	0.0001	0.4	128	12	max	1.	---	1.
9	0.01	0.01	0.3	32	12	max	1.	1.	---
10	0.01	0.001	0.4	128	6	max	0.	---	0.
11	0.0001	0.0001	0.4	128	18	avg	1.	1.	---
12	0.001	0.01	0.4	32	6	max	0.	0.	---
						Average:	0.75	0.75	0.75

The influence of hyperparameter values on the model's behavior is presented in the following sections. Information on each hyperparameter is summarized through tables and figures, which are grouped by set of values and show the individual distribution for each of the 12 folds used in Single-MINI's Nested-LOOCV procedure.

5.1.1 Learning rate

A low learning rate of 0.0001 was found to have the best results on MINI-RGBD, possibly due to the small and unbalanced nature of its data (see Table 14). By taking small steps in learning, a model is able to neglect large weights that could potentially lead to overfitting and small sensitivity values. Indeed, along with the F1 score of 0.71, a sensitivity of 85% was achieved

using 0.0001, far from the 62% and 64% of $\alpha = 0.01$ and $\alpha = 0.001$, which varied significantly on different folds (see Figure 26).

Table 14 - Performance metrics for learning rate (α) values on Single-MINI.
Values throughout all (n=480) training runs on the Single-MINI-RGBD experimental setup.

α	Metric, avg. (std.)							Nº
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
0.01	0.64 (0.22)	0.75 (0.44)	0.54 (0.5)	0.51 (0.05)	0.51 (0.37)	0.59 (0.37)	0.53 (0.05)	169
0.001	0.62 (0.21)	0.73 (0.44)	0.51 (0.5)	0.51 (0.04)	0.48 (0.35)	0.57 (0.35)	0.52 (0.04)	154
0.0001	0.72 (0.24)	0.85 (0.36)	0.59 (0.5)	0.52 (0.05)	0.64 (0.35)	**0.71 (0.34)**	0.53 (0.06)	157

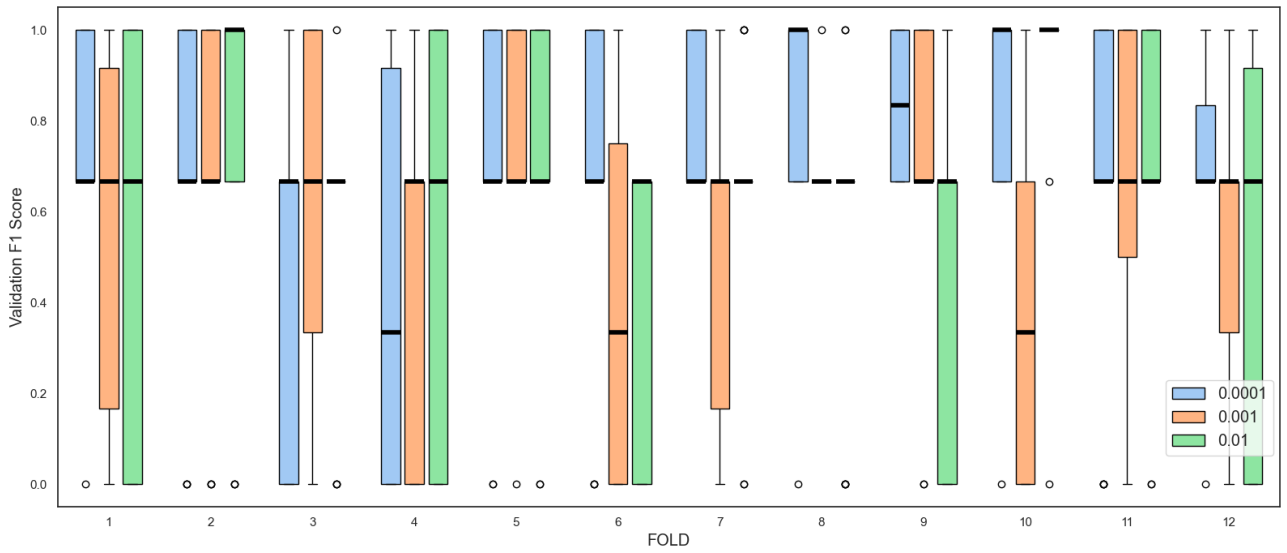


Figure 26 - Distribution of the F1 Score for different learning rate (α) values on Single-MINI.

5.1.2 Weight decay

The same can be loosely said about weight decay. "Large" weight decay values, such as $\lambda = 0.0001$ seem to have had an impact on how large weights – in this case, due mostly to negative classes – affect the model. It can be seen that $\lambda = 0.01$, although having a lower average of the F1 score, has a more balanced trade-off between sensitivity and specificity (see Table 15 and Figure 27). In our particular application, however, it is more desirable that the classification system be able to correctly classify positive instances, even if it increases the number of false

negatives. This is the case because it is often better to initiate GMA-related treatment (e.g., physiotherapy) on healthy infants than it is to discharge unhealthy infants. Thus, we see $\lambda = 0.001$ and $\lambda = 0.0001$ as better performing models.

Table 15 - Performance metrics for weight decay (λ) values on Single-MINI.
Values throughout all (n=480) training runs on the Single-MINI experimental setup.

λ	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
0.01	0.66 (0.23)	0.71 (0.45)	0.6 (0.49)	0.51 (0.05)	0.52 (0.39)	0.58 (0.4)	0.53 (0.06)	164
0.001	0.66 (0.23)	0.81 (0.39)	0.5 (0.5)	0.51 (0.05)	0.56 (0.35)	**0.64 (0.35)**	0.53 (0.06)	153
0.0001	0.66 (0.23)	0.8 (0.4)	0.52 (0.5)	0.51 (0.05)	0.56 (0.36)	**0.64 (0.35)**	0.53 (0.06)	163

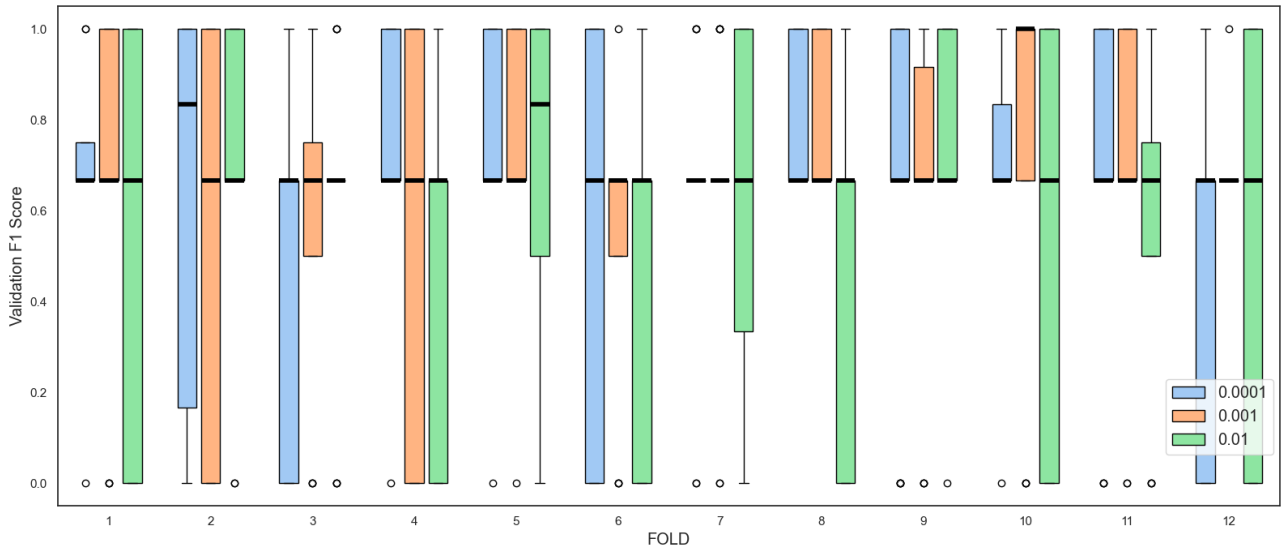


Figure 27 - Distribution of the F1 Score for different weight decay (λ) values on Single-MINI.

5.1.3 Threshold

The best performing model's threshold value was 0.3, rendering an F1 score of 0.67 and precision of 0.6 (see Table 16 and Figure 28). A value of 0.3 meant that 30% of the bins had to be predicted as positive to ascribe a positive class to the sequence. Thus, it is clear that a small value would increase sensitivity, but not so that it would also increase specificity (58%). The

reason this happened is not clear as far as this analysis reaches, but correlation analysis between different hyperparameters could give insights into it.

Table 16 - Performance metrics for threshold ($thr.$) values on Single-MINI experimental setup.
Values throughout all (n=480) training runs on the Single-MINI experimental setup.

thr.	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
0.3	0.7 (0.24)	0.81 (0.39)	0.58 (0.49)	0.52 (0.05)	0.6 (0.37)	**0.67 (0.36)**	0.53 (0.06)	153
0.4	0.65 (0.23)	0.77 (0.43)	0.54 (0.5)	0.51 (0.05)	0.53 (0.37)	0.61 (0.37)	0.53 (0.06)	162
0.5	0.63 (0.22)	0.75 (0.43)	0.52 (0.5)	0.51 (0.05)	0.51 (0.36)	0.59 (0.37)	0.52 (0.05)	165

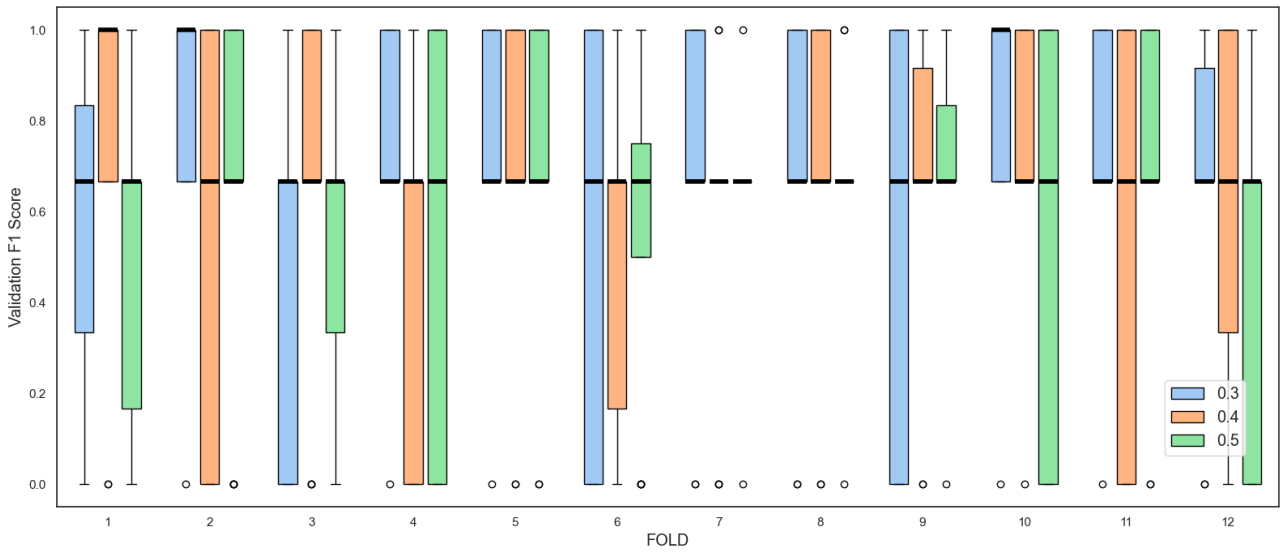


Figure 28 - Distribution of the F1 Score for different threshold ($thr.$) values on Single-MINI.

5.1.4 Number of feature maps and number of bins

Few conclusions can be made from the distribution of F1 scores for number of feature maps and bins during optimization. Small differences in their means and standard deviations suggest these can have less impact on the model predictions than other hyperparameters (see Table 17 and Table 18). The mean sensitivity of $\mathbf{ft_m} = 32$ shows a slight improve in detecting positives, backed up by the concurrent increase in precision. Overall, Figure 29 and Figure 30 illustrate

that different values for $bins$ and $\mathbf{ft_m}$ vary greatly on different folds and have inconsistent behavior.

Table 17 - Performance metrics for nº of feature maps ($\mathbf{ft_m}$) values on Single-MINI.
Values throughout all (n=480) training runs on the Single-MINI experimental setup.

$\mathbf{ft_m}$	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
32	0.69 (0.24)	0.81 (0.4)	0.57 (0.5)	0.52 (0.05)	0.59 (0.37)	**0.66 (0.36)**	0.53 (0.06)	144
64	0.63 (0.22)	0.75 (0.44)	0.52 (0.5)	0.51 (0.05)	0.51 (0.36)	0.59 (0.37)	0.52 (0.06)	159
128	0.66 (0.23)	0.77 (0.42)	0.55 (0.5)	0.51 (0.05)	0.55 (0.37)	0.62 (0.37)	0.53 (0.06)	177

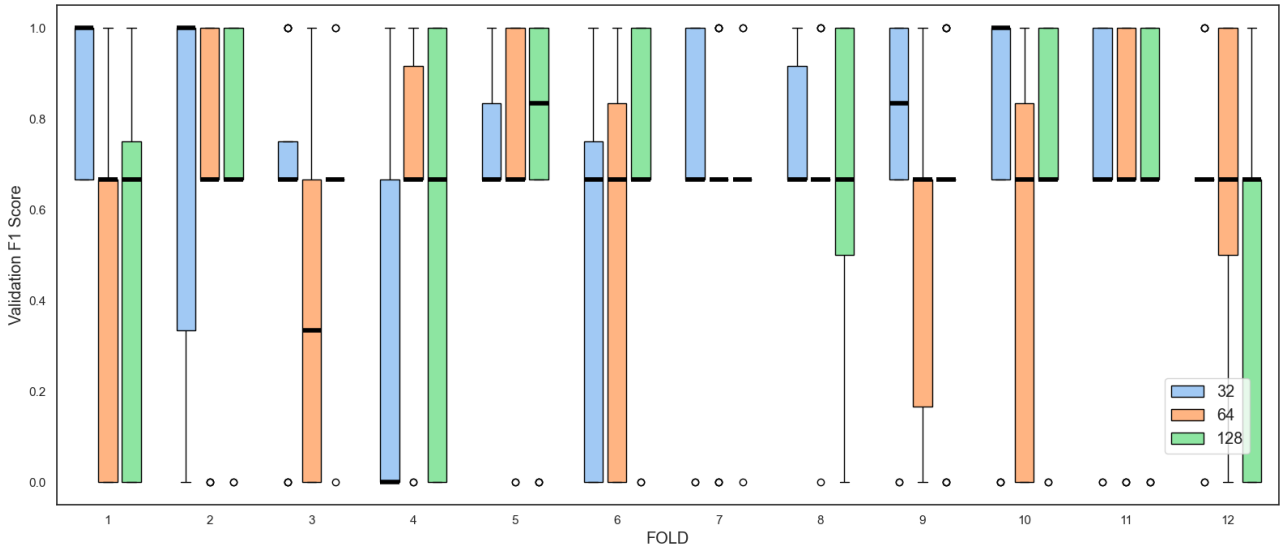


Figure 29 - Distribution of the F1 Score for different nº of feature maps ($\mathbf{ft_m}$) values on Single-MINI.

Table 18 - Performance metrics for nº of bins ($bins$) values on Single-MINI.
Values throughout all (n=480) training runs on the Single-MINI experimental setup.

$bins$	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
6	0.65 (0.23)	0.78 (0.42)	0.51 (0.5)	0.51 (0.05)	0.53 (0.36)	0.62 (0.36)	0.52 (0.06)	162
12	0.64 (0.23)	0.79 (0.41)	0.49 (0.5)	0.51 (0.06)	0.54 (0.35)	**0.63 (0.35)**	0.52 (0.07)	170
18	0.69 (0.24)	0.75 (0.43)	0.64 (0.48)	0.51 (0.04)	0.57 (0.39)	**0.63 (0.39)**	0.53 (0.05)	148

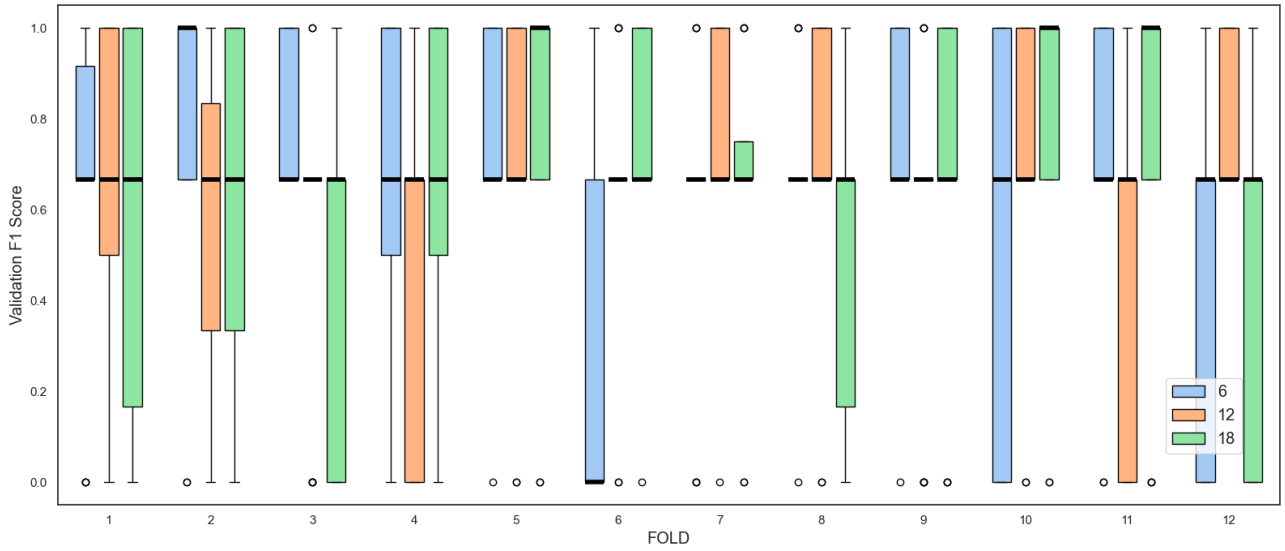


Figure 30 - Distribution of the F1 Score for different nº of bins ($bins$) values on Single-MINI.

5.1.5 Pooling methods

Pooling had a significant effect on classification. Table 19 shows an F1 score of 0.81 for the max pooling method, compared to 0.43 for mean pooling. Additionally, a standard deviation of 0.22 and the distributions in Figure 31 describe a consistent performance for this hyperparameter. It can also be seen that the F1 score median for 5 out of the 12 folds was equal to $1.$ and no value below 0.6 occurred.

Table 19 - Performance metrics for type of pooling method ($pool$) on Single-MINI.
Values throughout all (n=480) training runs on the Single-MINI experimental setup.

pool	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
max	0.74 (0.25)	0.97 (0.17)	0.51 (0.5)	0.52 (0.06)	0.73 (0.28)	**0.81 (0.22)**	0.54 (0.07)	243
mean	0.58 (0.18)	0.57 (0.5)	0.58 (0.49)	0.5 (0.03)	0.36 (0.36)	0.43 (0.39)	0.52 (0.04)	237

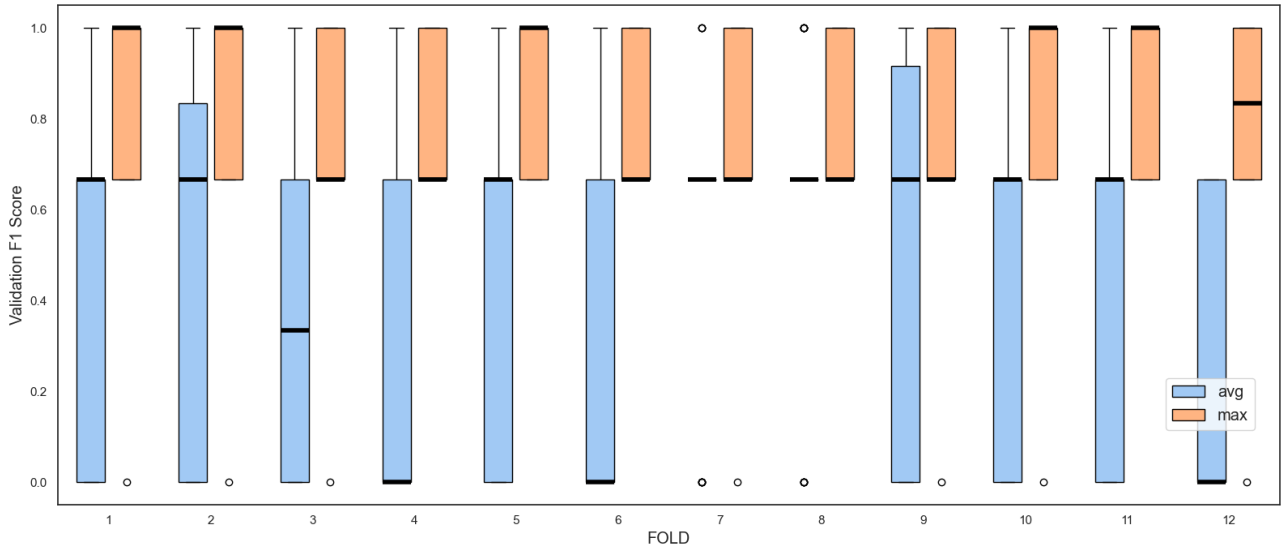


Figure 31 - Distribution of the F1 Score for different methods of pooling ($pool$) on Single-MINI.

5.2 Single-RVI

For this experiment, we trained on 80% of the whole dataset, 20% of which were used to validate and perform hyperparameter optimization, while testing was done on the remaining 20%.

The dataset consists originally of 38 samples, 6 of which were positive classes. As a data augmentation step, we segmented sequences into smaller, 1000 frames-wide (see Section 4.2.2), thus resulting in 124 samples with an equal class distribution.

Each split had approximately 20% of positive classes. Ten iterations were run so we could gather summary statistics of both validation and testing splits. Our optimal models do not reach state-of-the-art models such as those in McCay *et al*'s (2022) study, which yielded an accuracy of 97.37%. While their model reached 100% specificity and had trouble classifying all positive classes, ours achieved 100% sensitivity, and some of our models reached 100% accuracy. Despite average results on the validation splits during optimization, the optimal models found throughout the experiment yielded decent results on the testing splits, with a permanent sensitivity of 100% and 5 out of 10 iterations with 100% accuracy. The remaining 5 had clear issues to predict negative classes, as they might have overfitted to positive samples' features. See Table 20 for the details of the optimal models and their performance on Single-RVI's 10 iterations.

The influence of hyperparameter values on the model's behavior is presented in the following sections. Figures and Tables are here reported in the same manner as in the previous section.

Table 20 - Optimal model for each of the 10 iterations on Single-RVI and their performance on the testing split.

Iteration	Hyperparameter values						Performance on testing split (n=5)			
	α	λ	*thr.*	ft_m	*b*	*pool*	Acc.	Sns.	Spc.	F1
1	0.001	0.0001	0.5	128	12	max	1.	1.	1.	1.
2	0.001	0.0001	0.5	128	12	max	1.	1.	1.	1.
3	0.001	0.0001	0.5	128	12	max	1.	1.	1.	1.
4	0.0001	0.01	0.3	64	18	max	1.	1.	1.	1.
5	0.0001	0.01	0.3	64	18	max	0.2	1.	0.	0.33
6	0.0001	0.01	0.3	64	18	max	0.4	1.	0.25	0.4
7	0.0001	0.01	0.3	64	18	max	1.	1.	1.	1.
8	0.0001	0.01	0.3	64	18	max	0.4	1.	0.25	0.4
9	0.0001	0.01	0.3	64	18	max	0.6	1.	0.5	0.5
10	0.0001	0.01	0.3	64	18	max	0.4	1.	0.25	0.4
						Average:	0.7	1.	0.625	0.703

5.2.1 Learning rate

The intermediate value $\alpha = 0.001$ had the best performance as it yielded balanced values for sensitivity and specificity. Although the model skews to the negative, more populated, class, with a specificity of 82%, different learning rates are able to reduce this skewness and predict positive classes more consistently, as seen by the precision of 79% of $\alpha = 0.001$, compared to 61% and 60% on Table 21. Moreover, Figure 32 shows that the averages for this value of α are consistently higher and with less poor performing cases than, for instance, $\alpha = 0.01$.

Table 21 - Performance metrics for learning rate (α) values on Single-RVI.
Values throughout all (n=400) training runs on the Single-RVI experimental setup.

α	Metric, avg. (std.)							Nº
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
0.01	0.79 (0.2)	0.58 (0.26)	0.82 (0.26)	0.54 (0.04)	0.6 (0.32)	0.49 (0.15)	0.18 (0.03)	121
0.001	0.86 (0.16)	0.57 (0.25)	0.91 (0.21)	0.58 (0.05)	0.79 (0.28)	**0.58 (0.16)**	0.21 (0.05)	130
0.0001	0.81 (0.18)	0.56 (0.26)	0.85 (0.24)	0.55 (0.05)	0.61 (0.33)	0.5 (0.18)	0.19 (0.04)	149

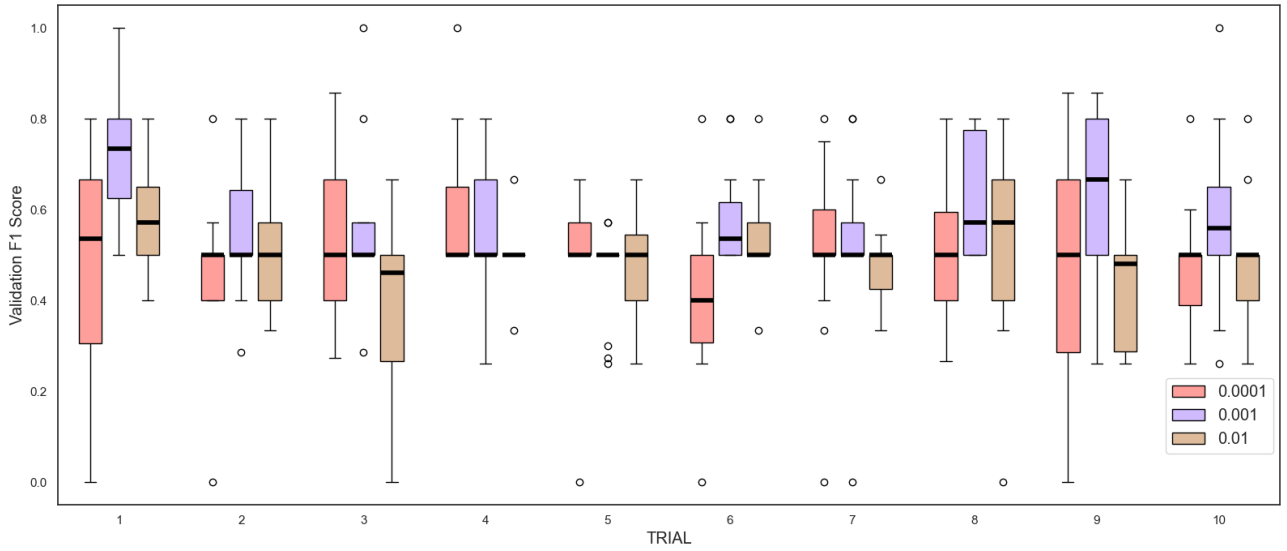


Figure 32 - Distribution of the F1 Score for different learning rate (α) values on Single-RVI.

5.2.2 Weight decay

Weight decay had a lesser impact on all trials, with no apparent significance. F1 score's medians for all values throughout the trials are around 0.5, as seen in Figure 33. Regarding outliers, $\lambda = 0.0001$ had more cases of $F1 = 1$, which might suggest that "larger" decays help in reducing overfitting and improving generalization – as happened with Single-MINI. All metrics, means and standard deviations, for weight decay are similar (see Table 22).

Table 22 - Performance metrics for weight decay (λ) values on Single-RVI.
Values throughout all (n=400) training runs on the Single-RVI experimental setup.

λ	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
0.01	0.81 (0.19)	0.56 (0.27)	0.85 (0.25)	0.55 (0.05)	0.64 (0.33)	0.5 (0.18)	0.19 (0.04)	144
0.001	0.82 (0.17)	0.58 (0.24)	0.86 (0.22)	0.56 (0.05)	0.68 (0.31)	**0.53 (0.15)**	0.19 (0.04)	126
0.0001	0.82 (0.18)	0.57 (0.26)	0.87 (0.24)	0.56 (0.06)	0.68 (0.32)	**0.53 (0.18)**	0.2 (0.05)	130

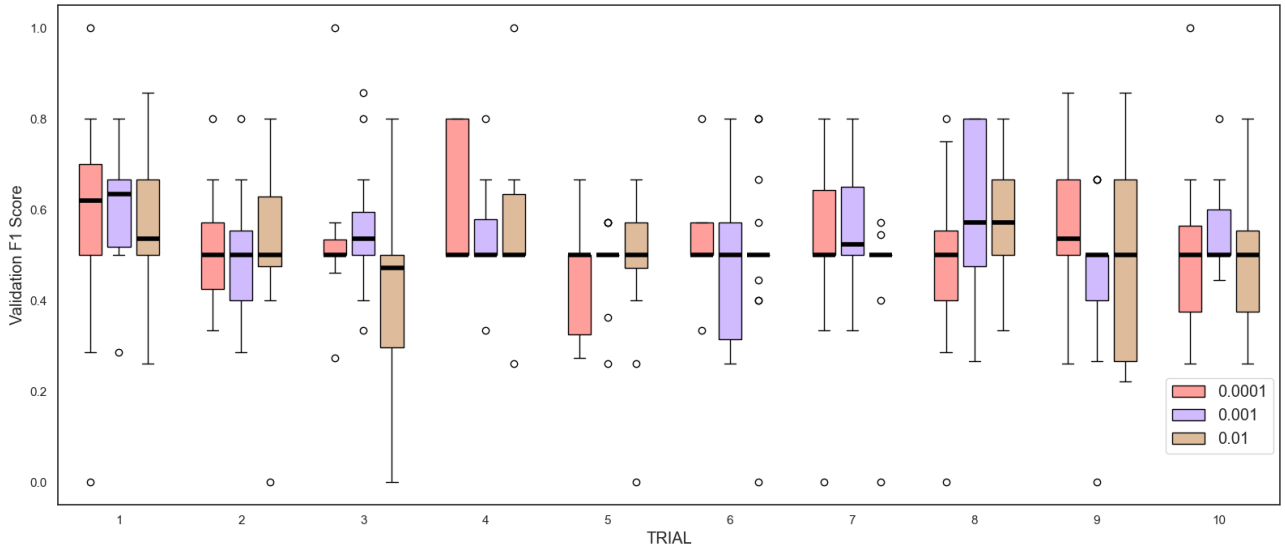


Figure 33 - Distribution of the F1 Score for different weight decay (λ) values on Single-RVI.

5.2.3 Threshold

Small threshold values show a better results, slightly improving sensitivity (see Table 23). Figure 34 shows that median value for $thr. = 0.3$ are mostly higher, withs some trials obtaining high F1 scores. Regarding other values, 0.4 and 0.5 both show similar performance and distributions of the F1 score, although models with $thr. = 0.5$ had instances of $F1 = 1$ inside their standard deviations (see Figure 34). Again, correlation analysis could show whether these values are due to other hyperparameters co-occurrences.

Table 23 - Performance metrics for threshold ($thr.$) values on Single-RVI.
Values throughout all (n=400) training runs on the Single-RVI experimental setup.

thr.	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
0.3	0.84 (0.16)	0.6 (0.25)	0.88 (0.21)	0.57 (0.05)	0.7 (0.3)	**0.56 (0.17)**	0.2 (0.05)	118
0.4	0.81 (0.19)	0.57 (0.25)	0.85 (0.25)	0.55 (0.05)	0.63 (0.32)	0.51 (0.16)	0.19 (0.04)	123
0.5	0.81 (0.2)	0.55 (0.26)	0.85 (0.26)	0.55 (0.05)	0.67 (0.33)	0.5 (0.18)	0.19 (0.04)	159

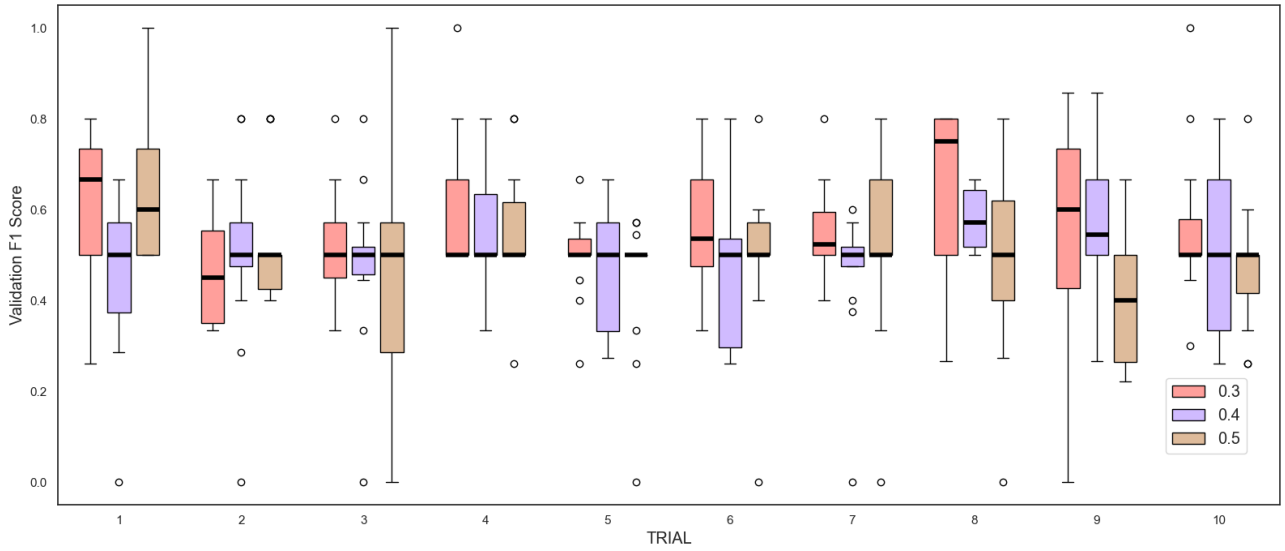


Figure 34 - Distribution of the F1 Score for different threshold ($thr.$) values on Single-RVI.

5.2.4 Number of feature maps and number of bins

A higher value for precision was found using 64 feature maps: 71% compared to $63/66\%$ on $\mathbf{ft_m} = \mathbf{32}, = \mathbf{128}$, respectively (see Table 24). Means for this number of feature maps fluctuated slightly above others, and had more values inside their upper quartile, as seen in Figure 35. Regarding number of bins, no significant impact was found on different trials (see Table 26 and Figure 36).

Table 24 - Performance metrics for nº of feature maps ($\mathbf{ft_m}$) values on Single-RVI.
Values throughout all (n=400) training runs on the Single-RVI experimental setup.

$\mathbf{ft_m}$	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
32	0.79 (0.22)	0.57 (0.27)	0.83 (0.28)	0.55 (0.05)	0.63 (0.34)	0.5 (0.19)	0.19 (0.04)	122
64	0.84 (0.16)	0.55 (0.26)	0.89 (0.21)	0.56 (0.05)	0.71 (0.32)	**0.53 (0.17)**	0.19 (0.05)	123
128	0.82 (0.17)	0.58 (0.25)	0.86 (0.22)	0.56 (0.05)	0.66 (0.31)	**0.53 (0.15)**	0.2 (0.04)	155

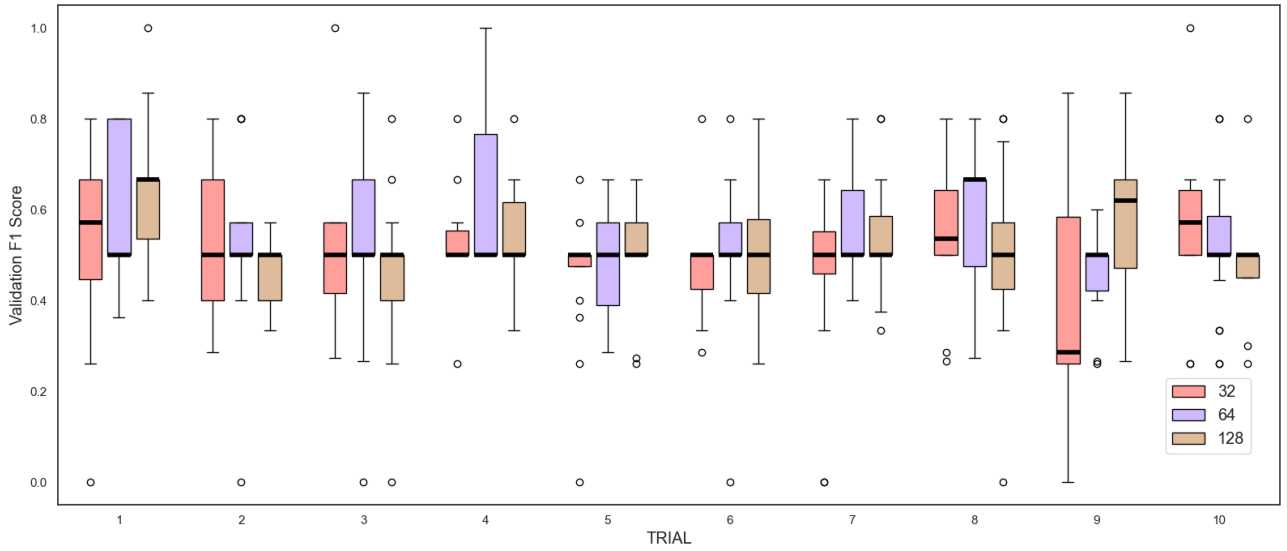


Figure 35 - Distribution of the F1 Score for different nº of feature maps ($\mathbf{ft_m}$) values on Single-RVI.

Table 25 - Performance metrics for nº of bins (*bins*) values on Single-RVI.
Values throughout all (n=400) training runs on the Single-RVI experimental setup.

bins	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
6	0.8 (0.21)	0.58 (0.27)	0.83 (0.27)	0.55 (0.05)	0.64 (0.32)	0.51 (0.16)	0.19 (0.04)	136
12	0.83 (0.16)	0.58 (0.26)	0.88 (0.21)	0.56 (0.06)	0.67 (0.32)	**0.53 (0.18)**	0.2 (0.05)	138
18	0.82 (0.18)	0.55 (0.24)	0.87 (0.24)	0.56 (0.05)	0.68 (0.32)	0.52 (0.18)	0.19 (0.04)	126

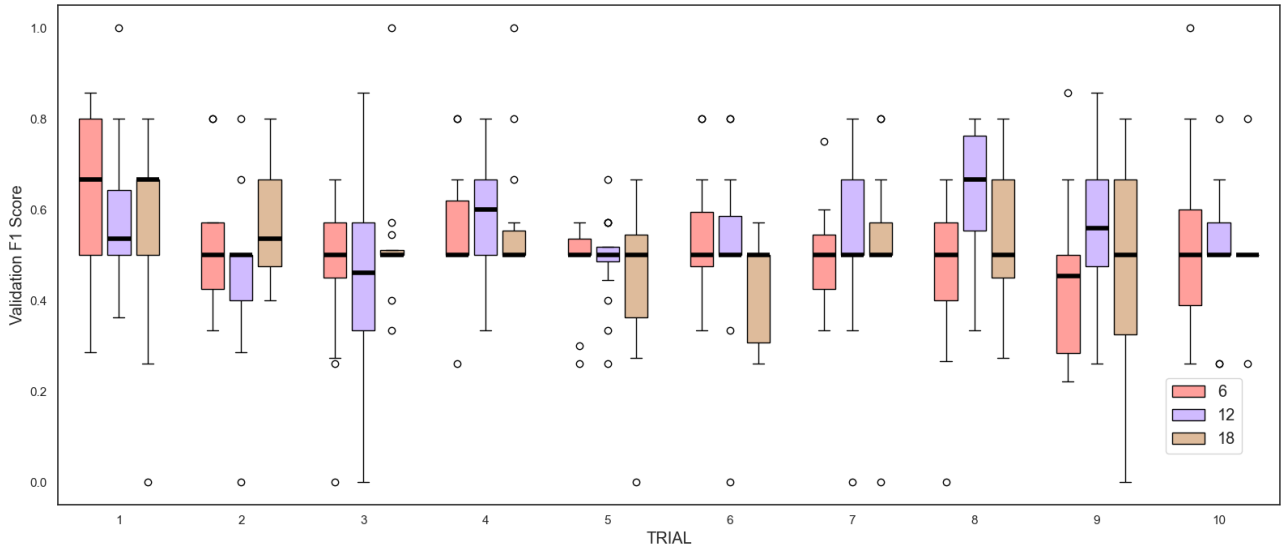


Figure 36 - Distribution of the F1 Score for different nº of bins (*bins*) values on Single-RVI.

5.2.5 Pooling methods

Finally, differently from what was found on Single-MINI, pooling methods had a lesser impact on Single-RVI. Max pooling increased sensitivity but dropped precision values, meaning that it overall predicted more positive classes, not necessarily correctly. A higher F1 score, thus, was achieved by mean pooling (see Table 26). Figure 37 shows that max pooling, yet, had more outliers in the higher end of F1 scores.

Table 26 - Performance metrics for type of pooling method ($pool$) on Single-RVI.
Values throughout all (n=400) training runs on the Single-RVI experimental setup.

$pool$	Metric, avg. (std.)							Nº
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
max	0.77 (0.23)	0.63 (0.26)	0.8 (0.3)	0.55 (0.06)	0.61 (0.34)	0.51 (0.19)	0.18 (0.03)	193
mean	0.86 (0.1)	0.52 (0.24)	0.92 (0.14)	0.56 (0.05)	0.71 (0.3)	**0.53 (0.15)**	0.21 (0.05)	207

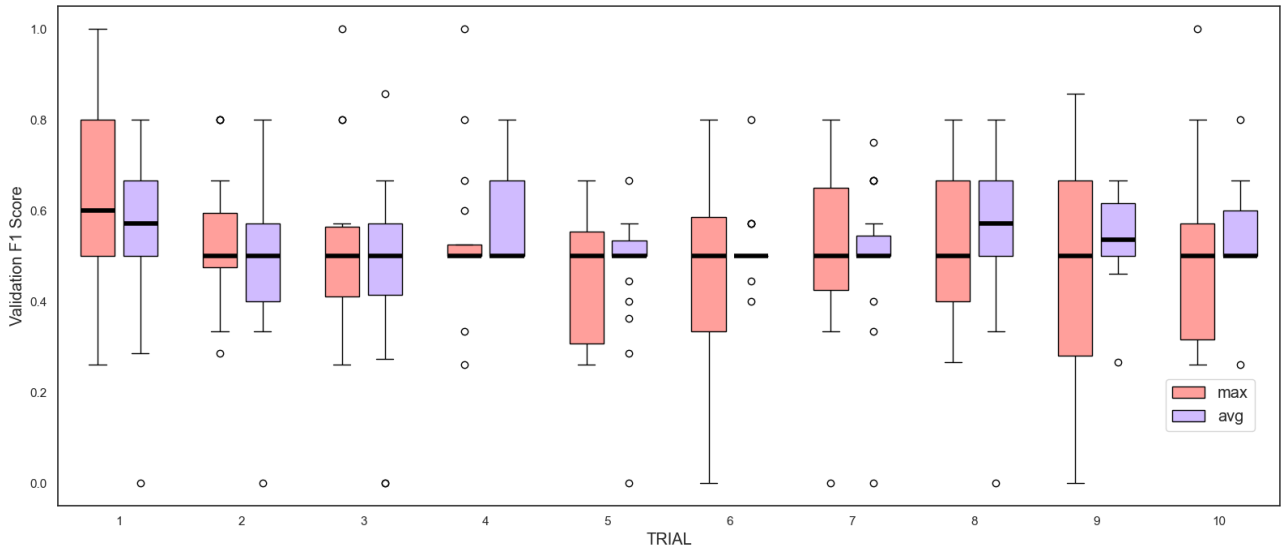


Figure 37 - Distribution of the F1 Score for different methods of pooling ($pool$) on Single-RVI.

5.3 Single-PMI

Among the available datasets, the PMI-GMA outstands in that it has a balanced, near 50/50%, ratio of negative and positive classes. In total, it contains 1120 segments, which were separated into different splits. This experiment follows the same procedure as Single-RVI's: 80% of data goes into training and 20% is used for testing; 20% of the training split is used for validation during hyperparameter optimization.

Due to the PMI-GMA size, only 5 iterations were run. Among the Single experiments, our model had the poorest performance with the PMI-GMA dataset, with optimal models achieving an average accuracy of 51%, near to a random classifier. Table 27 shows that models either overfitted to positive or negatives classes, generating sensitivity/specificity combinations of 97/3% and 0/100%, for instance. In fact, all optimal models had a precision value of approximately 50%, meaning that it ascribed one class to mostly all samples.

Table 27 - Optimal model for each of the 5 iterations on Single-PMI and their performance on the testing split.

Iteration	Hyperparameter values						Performance on testing split (n=45)			
	α	λ	*thr.*	ft_m	*b*	*pool*	Acc.	Sns.	Spc.	F1
1	0.0001	0.001	0.5	32	18	max	0.5	0.97	0.03	0.66
2	0.0001	0.001	0.5	32	18	max	0.52	0.98	0.05	0.67
3	0.01	0.0001	0.5	128	6	max	0.5	0.	1.	0.
4	0.0001	0.01	0.5	64	6	avg	0.52	0.9	0.14	0.6
5	0.0001	0.01	0.5	64	6	avg	0.52	0.83	0.21	0.33
						Average:	0.51	0.74	0.28	0.45

Driven by the poor performance on the PMI-GMA, we do not report our hyperparameter optimization process for Single-PMI, as it seemed to rely mostly on chance and no parameter had a significant impact on performance. We still evaluate the effect of PMI's data on our next experiment, containing data from all available datasets.

5.4 MINI+RVI+PMI

All samples (n=1256) were used for this experiment. By joining data from different datasets, we aim to assess our model's generalization power. We do a (80/20%)/20% split for training, validation, and testing, and perform 5 training iterations with 40 models containing randomly selected hyperparameters.

As we are, to our knowledge, novel to include all publicly available GM-annotated datasets, we are not able to compare our results externally. Additionally, despite the pre-processing procedure described, it is not clear whether our assumptions regarding model architecture are optimal for this kind of setup.

Lastly, recall that MINI-RGBD's samples had a length of 1000 frames and RVI's were segmented to 1000 as an augmentation process. PMI's samples, however, are 300-frames wide. The impact of this has not been deeply investigated, and poor performance might arise from such differences.

Our optimal models in this experiment yielded an average accuracy of 62%, 83% sensitivity, and 44% specificity. Compared to the significantly poor performance on Single-PMI, these are average results above a random classifier. The model predicts most instances as positive, which might be due to the small threshold values experimented with. For this experiment, F1 Scores were mostly the same across *all* trials with *all* hyperparameters. Thus, we set specificity as our metric of interest. We also report model's training statistics.

Table 28 - Optimal model for each of the 6 iterations on MINI+RVI+PMI and their performance on the testing split.

Iteration	**Hyperparameter values**						**Performance on testing split (n=51)**			
	α	λ	*thr.*	ft_m	*b*	*pool*	Acc.	Sns.	Spc.	F1
1	0.0001	0.0001	0.4	128	18	max	0.64	0.76	0.54	0.67
2	0.01	0.01	0.5	128	6	avg	0.57	0.97	0.21	0.68
3	0.0001	0.001	0.4	128	18	avg	0.64	0.79	0.5	0.67
4	0.01	0.0001	0.5	128	12	avg	0.63	0.8	0.48	0.67
5	0.01	0.0001	0.5	32	12	avg	0.63	0.82	0.45	0.67
						Average:	0.62	0.83	0.44	0.67

This experiment strongly indicates that those changes made in regularization and inner architecture of our model had a negative impact when dealing with a balanced dataset. Differently from the previous Single-MINI and Single-RVI (which were originally meant as the sole experiments), Single-PMI and MINI+RVI+PMI have a nearly balanced number of positives and negatives classes, and it is possible that the 90%> sensitivities occurred due to these decisions. Yet, as Table 28 shows, testing splits using the optimal models still managed to achieve 40-50% specificity values while maintaining $\pm 80\%$ sensitivity.

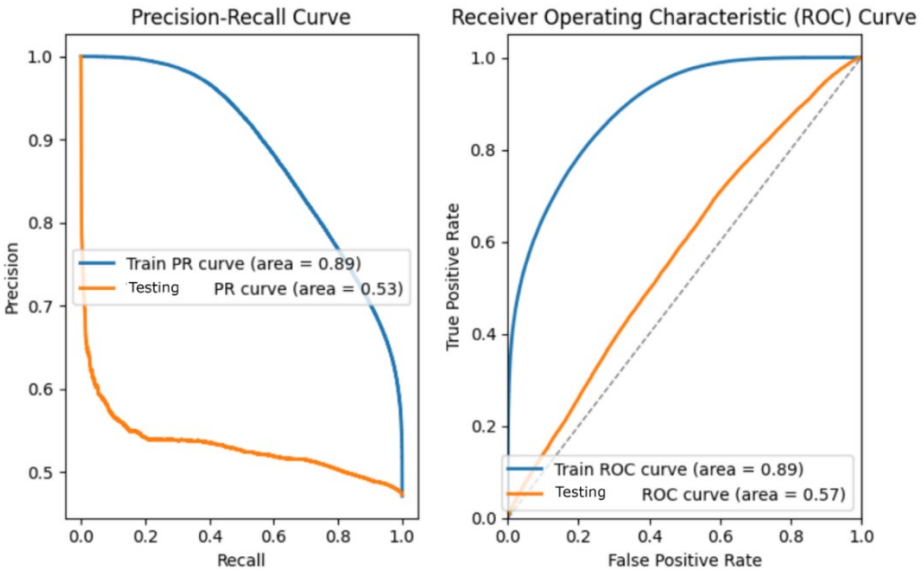


Figure 38 - PR and ROC curves for optimal model of iteration 1 during training and testing.

5.4.1 Learning rate

Small learning rates continue to be better for our small-sized training splits (see Table 29). By limiting learning steps, the model avoids large, often non-optimal, learning patterns. Although all values for α give high sensitivities, we consider $\alpha = 0.0001$ to be optimal given the resulting (more) balanced ratio of sensitivity and specificity. Figure 39 shows that a large α (in trial 1 and 3) might invert the model's predictions from mostly positive to negative, represented by the $\pm 100\%$ specificity outliers.

Table 29 - Performance metrics for different learning rate (α) values on MINI+RVI+PMI.
Values throughout all (n=200) training runs on the MINI+RVI+PMI experimental setup.

α	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
0.01	0.51 (0.02)	0.94 (0.2)	0.08 (0.21)	0.5 (0.01)	0.5 (0.09)	0.64 (0.12)	0.51 (0.01)	71
0.001	0.53 (0.02)	0.95 (0.07)	0.1 (0.08)	0.51 (0.01)	0.52 (0.01)	0.67 (0.02)	0.51 (0.01)	56
0.0001	0.53 (0.03)	0.9 (0.14)	**0.15 (0.16)**	0.5 (0.01)	0.52 (0.02)	0.65 (0.05)	0.5 (0.01)	73

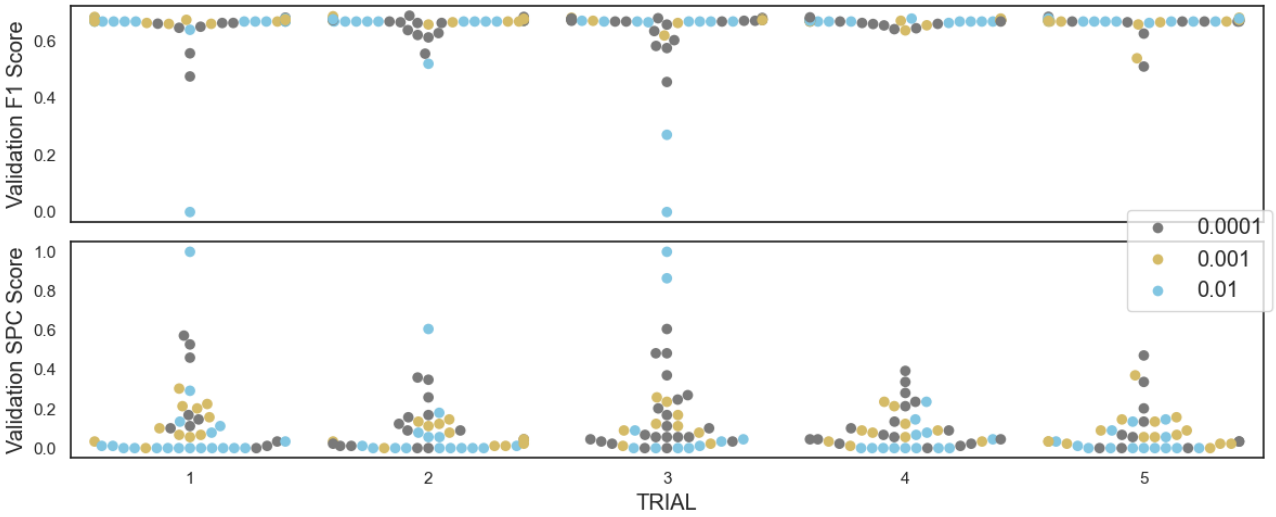


Figure 39 - Distribution of the F1 Score and specificity for different learning rate (α) values on MINI+RVI+PMI.

5.4.2 Weight decay

Weight decays, on the other hand, present an unusual behavior. A "small" value of $\lambda = 0.01$ was correlated with the higher – despite still low – specificity of 14% (see Table 30). In this case, it is plausible that our architectural choices regarding regularization, which aimed to counter-balance few positive classes in the data, did the opposite when dealing with this experiment which had a 50/50% positive/negative ratio of classes. Figure 40 shows for trial 3, for instance, that higher specificities were only obtained with the co-occurrence of low F1 scores and, therefore, low sensitivities.

Table 30 - Performance metrics for different weight decay (λ) values on MINI+RVI+PMI.
Values throughout all (n=200) training runs on the MINI+RVI+PMI experimental setup.

λ	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
0.01	0.53 (0.02)	0.9 (0.2)	**0.14 (0.21)**	0.5 (0.01)	0.5 (0.09)	0.64 (0.12)	0.51 (0.01)	76
0.001	0.52 (0.02)	0.94 (0.11)	0.09 (0.12)	0.5 (0.01)	0.51 (0.01)	0.66 (0.03)	0.51 (0.01)	62
0.0001	0.52 (0.02)	0.95 (0.12)	0.09 (0.13)	0.5 (0.01)	0.51 (0.01)	0.66 (0.04)	0.51 (0.01)	62

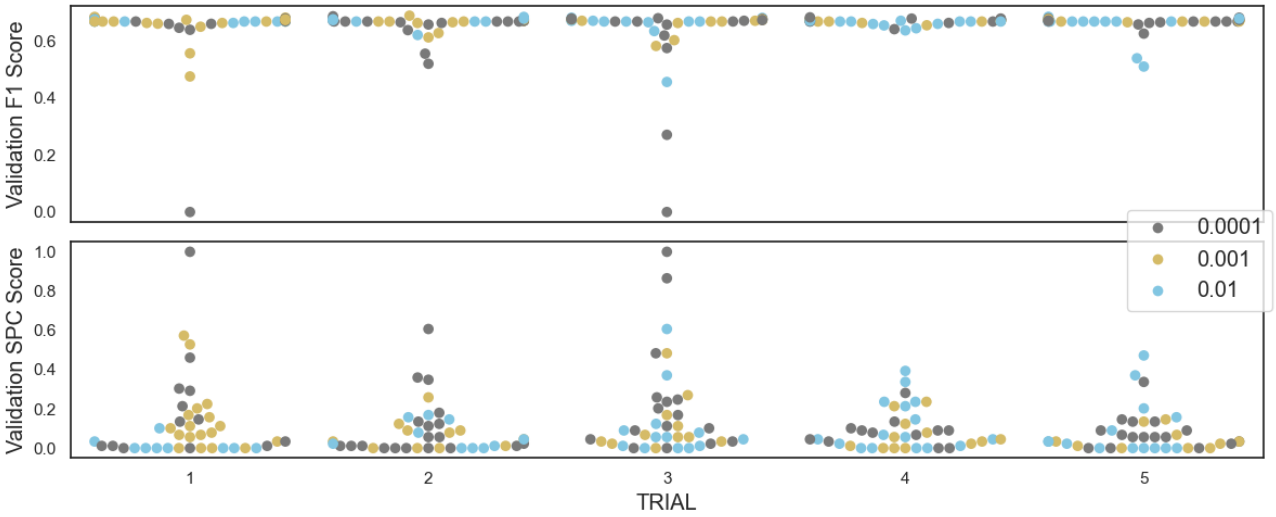


Figure 40 - Distribution of the F1 Score and specificity for different weight decay (λ) values on MINI+RVI+PMI.

5.4.3 Threshold

Figure 41 and Table 31 show that larger values for threshold work as intended, as a higher value $thr. = 0.5$ increases the specificity due to requiring more positive bins to be predicted as positive for a sample to be assigned the positive class. Accuracy was higher with lesser values, however, as the lower threshold values resulted in $93 - 98\%$ sensitivity rates with a not so significant (e.g. from 16% to 12%) change in specificity.

Table 31 - Performance metrics for different threshold ($thr.$) values on MINI+RVI+PMI.
Values throughout all (n=200) training runs on the MINI+RVI+PMI experimental setup.

thr.	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
0.3	0.53 (0.02)	0.93 (0.15)	0.12 (0.17)	0.51 (0.01)	0.52 (0.02)	0.66 (0.06)	0.51 (0.01)	69
0.4	0.52 (0.01)	0.98 (0.02)	0.05 (0.05)	0.5 (0.01)	0.51 (0.01)	0.67 (0.01)	0.5 (0.01)	64
0.5	0.52 (0.02)	0.88 (0.2)	**0.16 (0.21)**	0.5 (0.01)	0.5 (0.09)	0.63 (0.12)	0.51 (0.01)	67

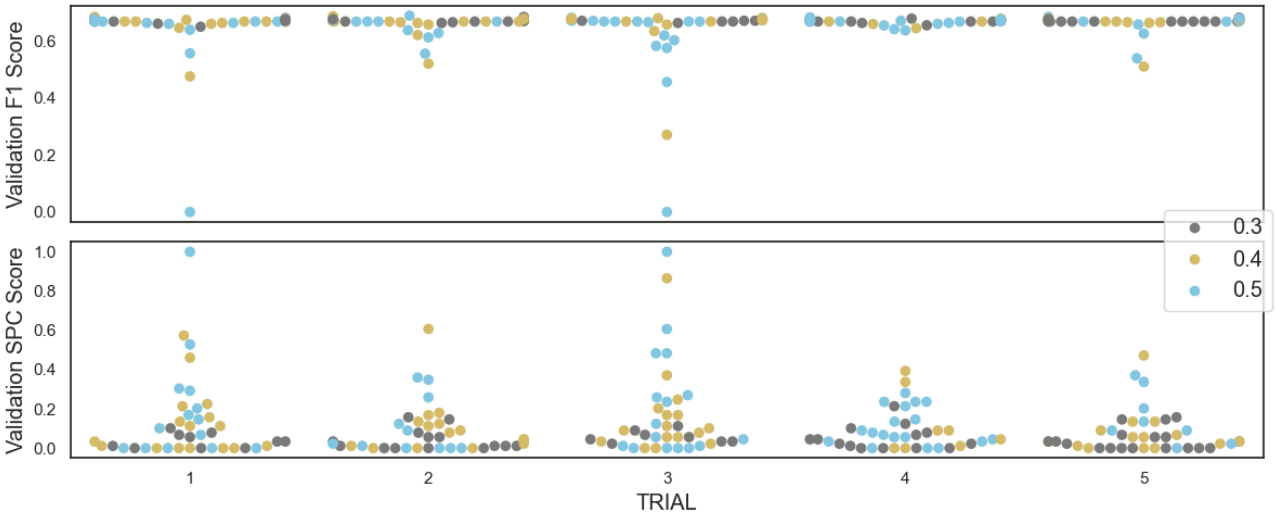


Figure 41 - Distribution of the F1 Score and specificity for different threshold ($thr.$) values on MINI+RVI+PMI.

5.4.4 Number of feature maps and number of bins

The number of feature maps and bins had little effect on changes of specificity, and seem to be insignificant for this experiment (see Figure 42/41 and Table 32Table 33/33).

Table 32 - Performance metrics for different nº of feature maps ($\mathbf{f_{mp}}$) values on MINI+RVI+PMI.
Values throughout all (n=200) training runs on the MINI+RVI+PMI experimental setup.

$\mathbf{f_{mp}}$	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
32	0.52 (0.02)	0.95 (0.11)	0.08 (0.13)	0.5 (0.01)	0.51 (0.01)	0.66 (0.04)	0.51 (0.01)	69
64	0.53 (0.03)	0.9 (0.19)	**0.15 (0.2)**	0.5 (0.01)	0.51 (0.09)	0.64 (0.12)	0.51 (0.01)	75
128	0.52 (0.02)	0.94 (0.12)	0.1 (0.13)	0.5 (0.01)	0.51 (0.01)	0.66 (0.04)	0.5 (0.01)	56

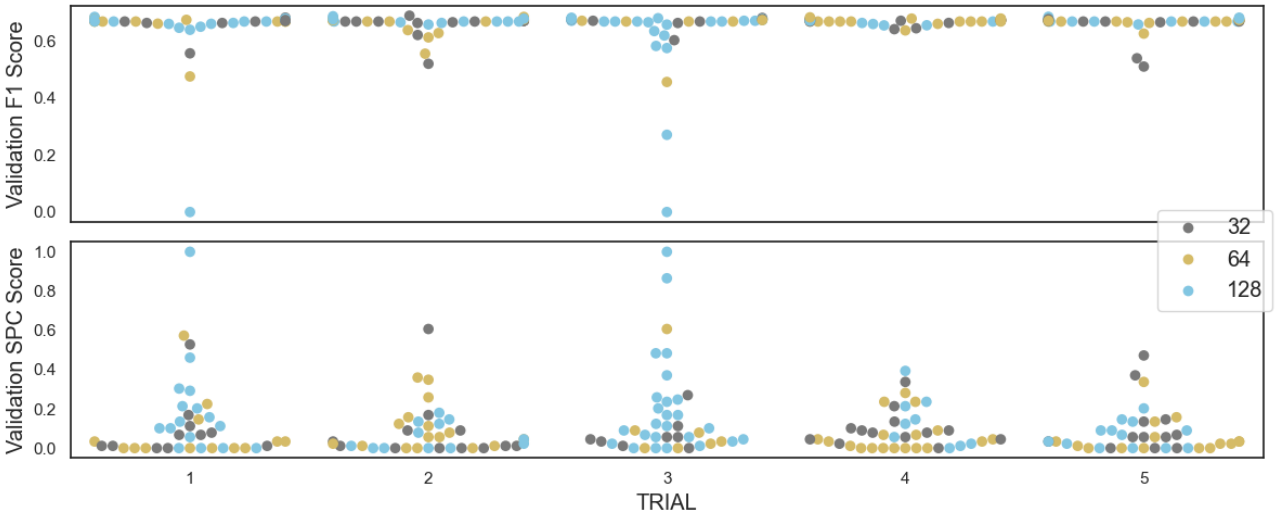


Figure 42 - Distribution of the F1 Score and specificity for different nº of feature maps ($\mathbf{f_{mp}}$) values on MINI+RVI+PMI.

Table 33 - Performance metrics for different nº of bins ($bins$) values on MINI+RVI+PMI.
Values throughout all (n=200) training runs on the MINI+RVI+PMI experimental setup.

bins	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
6	0.52 (0.02)	0.95 (0.12)	0.08 (0.14)	0.5 (0.01)	0.51 (0.01)	0.66 (0.05)	0.51 (0.01)	65
12	0.52 (0.02)	0.92 (0.16)	0.12 (0.17)	0.5 (0.01)	0.51 (0.06)	0.65 (0.09)	0.51 (0.01)	72
18	0.52 (0.02)	0.92 (0.17)	**0.13 (0.18)**	0.51 (0.01)	0.51 (0.07)	0.65 (0.09)	0.51 (0.01)	63

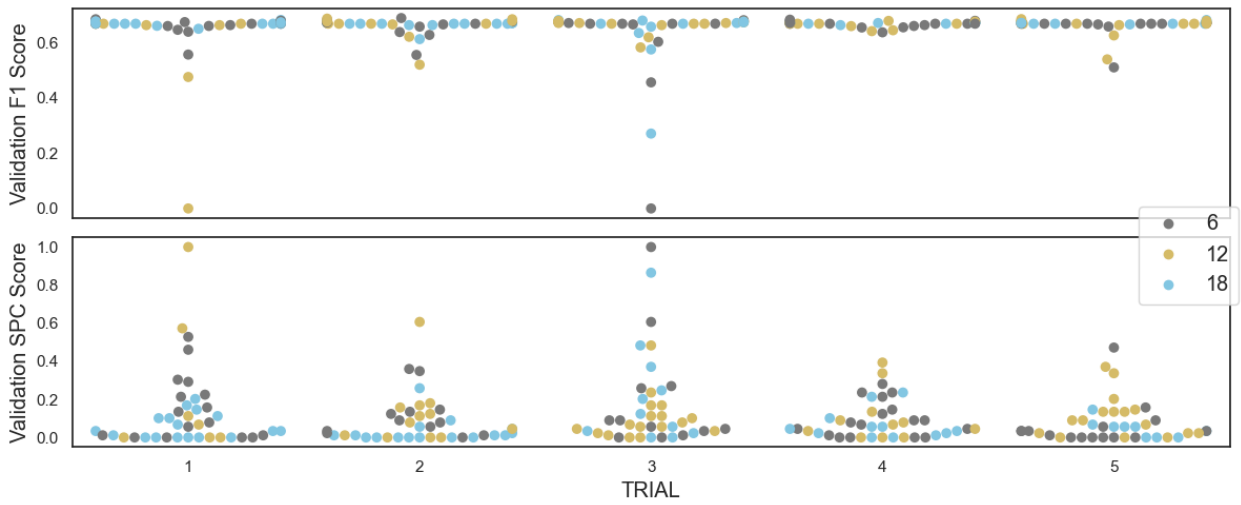


Figure 43 - Distribution of the F1 Score and specificity for different nº of bins ($bins$) values on MINI+RVI+PMI.

5.4.5 Pooling methods

Lastly, max pooling doubled the specificity of mean pooling, while retaining an F1 score of 0.65. Max pooling tends to be a better method than mean pooling as it focuses on the most prominent features, and not so on the overall behavior.

The overall behavior of both normal and abnormal GMs, as seen, can be easily mixed; the differences lie in the details. In this case, it can be supposed that mean pooling encouraged the model to overfit to the positive class, while max pooling reduced its influence – yielding a positive difference of 7% in sensitivity (see Table 34). No significant conclusions can be drawn from Figure 44.

Table 34 - Performance metrics for type of pooling method (*pool*) on MINI+RVI+PMI.
Values throughout all (n=200) training runs on the MINI+RVI+PMI experimental setup.

pool	**Metric, avg. (std.)**							**Nº**
	Accuracy	Sensitivity	Specificity	AUC-ROC	Precision	F1 Score	PR-AUC	
max	0.53 (0.02)	0.9 (0.18)	**0.15 (0.19)**	0.5 (0.01)	0.51 (0.07)	0.65 (0.1)	0.51 (0.01)	104
mean	0.51 (0.01)	0.96 (0.11)	0.07 (0.12)	0.5 (0.01)	0.51 (0.01)	0.66 (0.04)	0.51 (0.01)	96

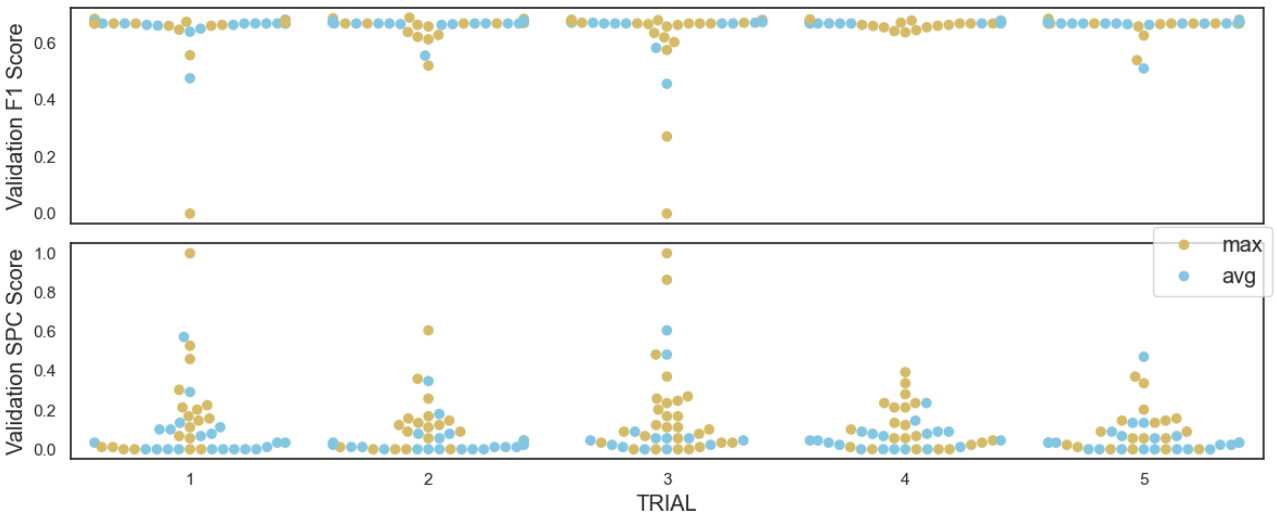


Figure 44 - Distribution of the F1 Score and specificity for different methods of pooling (*pool*) values on MINI+RVI+PMI.

6 CONCLUSION

This work theme has been the automation of the General Movements Assessment, a diagnostic tool for predicting neurodevelopmental risk on infants below 6 months of age. Gathering data from infants, especially video sequences, is a difficult process frequently hindered by ethical and anonymity issues. Recently, public datasets – available as the 2D pose coordinates of moving infants – have been published in the literature. By using these data, we explored how deep neural networks could be leveraged to classify sequences of infant movement.

6.1 Contributions

We proposed a throughout pre-processing pipeline so that different publicly available datasets, containing noisy and low-quality sequences, could be jointly used in experimental setups. To the best of our knowledge, we are the first work to report our results of both hyperparameter optimization and testing performance on data coming from multiple publicly datasets. By doing this, we are able to analyze how our model adapts when trying to generalize to more than one dataset's data. Besides, it encourages models not to overfit to features specific to a dataset – specifically small and unbalanced ones.

A great amount of data has been generated in our experiments, which could highly contribute to the discussion of GMA automation and how specific architectural choices relating to DNNs impact GM detection and classification. Additionally, a detailed set of metrics has been reported in order to transparently describe our model's performance.

6.2 Limitations

Our data is small-sized, consisting of only 1170 samples. Even when training shallow DNNs, the usual amount of data is much larger than ours, and more complex architectures tend to ask for more data. As our model is complex, it is expected that either overfitting or bad learning might occur. On top of that, the quality of our data – as well as overall hospital-derived data – is poor, which also contributes to the mentioned disadvantages. Furthermore, most of our pre-processing decisions are based on the related literature and have not been experimented with. For instance, although different interpolation techniques were qualitatively assessed via animations and exploratory analysis, their final impact on classification was not. The same can

be said about the process of computing features, window sliding, and histogram-encoding. Thus, although conceptually justified, these are unexperimented choices.

Our model architecture might not be best suited for the small-sized and poor-quality data, and could have been better optimized to deal with these characteristics. Weight decay, loss functions, and regularization steps were implemented, but additional modules and a reduced number of parameters could be experimented with.

Finally, our analysis of hyperparameter optimization is limited, and could be improved by many aspects. As mentioned throughout section 5.1., correlation analysis and significance analysis using non-parametric tests such as Krustal-Wallis would be beneficial for investigating hyperparameters more precisely. Regardless, we acknowledge our limited input data, and obtaining more data would most certainly allow for more robust and insightful discussion.

6.3 Future work

To mitigate the challenges posed by small datasets, promising data augmentations techniques are available. Variants of the synthetic minority over-sampling techniques (SMOTE) applied to skeleton-based human action recognition saw an increasing interest (Iglesias et al., 2023; Xin et al., 2023). The use of the Variational Autoencoder (VAE) architecture, for instance, have been explored for augmenting human motion sequences (Warchoł & Oszust, 2022). Self-occlusions are frequent in infant motion sequences and amount to a large portion of unusable signals outputted from pose estimation algorithms. It is interesting to explore algorithms for properly filling gaps resulting from occlusion, similar to interpolation, but specific to infants' body topology, such as Generative Adversarial Networks (GANs) (Saleh et al., 2023; Zhao et al., 2021).

Concerning architectural choices, different rationales might be followed. On one side, complex and innovative architectures such as Transformers and Attention-based modules should be experimented for their learning and generalization capabilities. On the other hand, the small-data nature of infant movement encourages and engineering of hand-made features able to capture GM patterns. While seemingly opposite, both ways should be led by an interest in the nature of general movements, and the design of either DNN architectures or quantitative translations of GM descriptions ask for more than just mere repetition.

REFERENCES

Adde, L., Helbostad, J., Jensenius, A. R., Langaas, M., & Støen, R. (Jan, 2013). Identification of fidgety movements and prediction of CP by the use of computer-based video analysis is more accurate when based on two video recordings. *Physiotherapy Theory and Practice*, *29*(6), 469–475. https://doi.org/10.3109/09593985.2012.757404

Adde, L., Yang, H., Sæther, R., Jensenius, A. R., Ihlen, E., Cao, J.-Y., & Støen, R. (Oct, 2018). Characteristics of general movements in preterm infants assessed by computer-based video analysis. *Physiotherapy Theory and Practice*, *34*(4), 286–292. https://doi.org/10.1080/09593985.2017.1391908

Ahmad, T., Jin, L., Zhang, X., Lai, S., Tang, G., & Lin, L. (Apr, 2021). Graph Convolutional Neural Network for Human Action Recognition: A Comprehensive Survey. *IEEE Transactions on Artificial Intelligence*, *2*(2), 128–145. https://doi.org/10.1109/TAI.2021.3076974

Akcakaya, N., Altunalan, T., Dogan, T., Yilmaz, A., & Yapici, Z. (Jan, 2019). Correlation of Prechtl Qualitative Assessment of General Movement Analysis with Neurological Evaluation: The Importance of Inspection in Infants. *Turkish Journal of Neurology*, *25*(2), 63–70. https://doi.org/10.4274/tnd.galenos.2018.98598

Akima, H. (Oct, 1970). A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures. *Journal of the ACM*, *17*(4), 589–602. https://doi.org/10.1145/321607.321609

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, … Farhan, L. (Mar, 2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53, 1–10. https://doi.org/10.1186/s40537-021-00444-8

Balasundaram, P., & Avulakunta, I. D. (2022). Bayley Scales Of Infant and Toddler Development. In *StatPearls*. StatPearls Publishing. http://www.ncbi.nlm.nih.gov/books/NBK567715/

Balta, D., Kuo, H., Wang, J., Porco, I., Morozova, O., Schladen, M., … Della Croce, U. (Sep, 2022). Characterization of Infants' General Movements Using a Commercial RGB-Depth Sensor and a Deep Neural Network Tracking Processing Tool: An Exploratory Study. *Sensors*, *22*(19), 1–11. https://doi.org/10.3390/s22197426

Bergstra, J., & Bengio, Y. (Feb, 2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, *13*(10), 281–305. https://jmlr.org/papers/v13/bergstra12a.html

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. http://gen.lib.rus.ec/book/index.php?md5=6b552b24cae380bb656f7aaef7f81b46

Bos, A. F., van Loon, A. J., Hadders-Algra, M., Martijn, A., Okken, A., & Prechtl, H. F. (Nov, 1997). Spontaneous motility in preterm, small-for-gestational age infants. II. Qualitative aspects. *Early Human Development*, *50*(1), 131–147. https://doi.org/10.1016/s0378-3782(97)00098-4

Bosanquet, M., Copeland, L., Ware, R., & Boyd, R. (Apr, 2013). A systematic review of tests to predict cerebral palsy in young children. *Developmental Medicine and Child Neurology*, *55*(5), 418–426. https://doi.org/10.1111/dmcn.12140

Caesar, R., Colditz, P. B., Cioni, G., & Boyd, R. N. (Nov, 2021). Clinical tools used in young infants born very preterm to predict motor and cognitive delay (not cerebral palsy): A systematic review. *Developmental Medicine and Child Neurology*, *63*(4), 387–395. https://doi.org/10.1111/dmcn.14730

Chambers, C., Seethapathi, N., Saluja, R., Loeb, H., Pierce, S., Bogen, D. … Kording, K. (Nov, 2020). Computer Vision to Automatically Assess Infant Neuromotor Risk. *IEEE Transactions on Neural*

Systems And Rehabilitation Engineering, *28*(11), 2431–2442. https://doi.org/10.1109/TNSRE.2020.3029121

Committee on Fetus and Newborn. (Nov, 2004). Age Terminology During the Perinatal Period. *Pediatrics*, *114*(5), 1362–1364. https://doi.org/10.1542/peds.2004-1915

Craciunoiu, O., & Holsti, L. (Jun, 2017). A Systematic Review of the Predictive Validity of Neurobehavioral Assessments During the Preterm Period. *Physical & Occupational Therapy In Pediatrics*, *37*(3), 292–307. https://doi.org/10.1080/01942638.2016.1185501

D. Sakkos, K. D. Mccay, C. Marcroft, N. D. Embleton, S. Chattopadhyay, & E. S. L. Ho. (Jan, 2021). Identification of Abnormal Movements in Infants: A Deep Neural Network for Body Part-Based Prediction of Cerebral Palsy. *IEEE Access*, *9*(1), 94281–94292. https://doi.org/10.1109/ACCESS.2021.3093469

de Vries, J. I., Visser, G. H., & Prechtl, H. F. (Dec, 1982). The emergence of fetal behaviour. I. Qualitative aspects. *Early Human Development*, *7*(4), 301–322. https://doi.org/10.1016/0378-3782(82)90033-0

Doroniewicz, I., Ledwoń, D. J., Affanasowicz, A., Kieszczyńska, K., Latos, D., Matyja, M., … Myśliwiec, A. (Oct, 2020). Writhing Movement Detection in Newborns on the Second and Third Day of Life Using Pose-Based Feature Machine Learning Classification. *Sensors*, *20*(21), 1–10. https://doi.org/10.3390/s20215986

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (Sep, 2022). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, *503*(1), 92–108. https://doi.org/10.1016/j.neucom.2022.06.111

Einspieler, C., Peharz, R., & Marschik, P. (May, 2016). Fidgety movements - Tiny in appearance, but huge in impact. *Jornal De Pediatria*, *92*(3), S64–S70. https://doi.org/10.1016/j.jped.2015.12.003

Einspieler, C., Prechtl, H. F., Bos, A. F., Ferrari, F., & Cioni, G. (2004). *Prechtl's Method on the Qualitative Assessment of General Movements in Preterm, Term and Young Infants*. Mac Keith Press.

Einspieler, C., Prechtl, H. F., Ferrari, F., Cioni, G., & Bos, A. F. (Nov, 1997). The qualitative assessment of general movements in preterm, term and young infants - Review of the methodology. *Early Human Development*, *50*(1), 47–60. https://doi.org/10.1016/s0378-3782(97)00092-3

Einspieler, C., & Prechtl, H. F. R. (Apr, 2005). Prechtl's assessment of general movements: A diagnostic tool for the functional assessment of the young nervous system. *Mental Retardation and Developmental Disabilities Research Reviews*, *11*(1), 61–67. https://doi.org/10.1002/mrdd.20051

Farnebäck, G. (2003). Two-Frame Motion Estimation Based on Polynomial Expansion. In *Image Analysis*, 363–370. Springer. https://doi.org/10.1007/3-540-45103-X_50

Ferrari, F., Cioni, G., & Prechtl, H. F. R. (Sep, 1990). Qualitative changes of general movements in preterm infants with brain lesions. *Early Human Development*, *23*(3), 193–231. https://doi.org/10.1016/0378-3782(90)90013-9

Filtjens, B., Vanrumste, B., & Slaets, P. (Dec, 2023). Skeleton-Based Action Segmentation with Multi-Stage Spatial-Temporal Graph Convolutional Neural Networks. *IEEE Transactions on Emerging Topics in Computing*, *12*(1), 202–212. https://doi.org/10.1109/TETC.2022.3230912

Fontana, C., Ottaviani, V., Veneroni, C., Sforza, S., Pesenti, N., Mosca, F., … Dellaca, R. (Aug, 2021). An Automated Approach for General Movement Assessment: A Pilot Study. *Frontiers In Pediatrics*, *9*(1), 1–10. https://doi.org/10.3389/fped.2021.720502

Fritsch, F. N., & Butland, J. (Sep, 1984). A Method for Constructing Local Monotone Piecewise Cubic Interpolants. *SIAM Journal on Scientific and Statistical Computing*, *5*(2), 300–304. https://doi.org/10.1137/0905021

Gao, Y., Long, Y., Guan, Y., Basu, A., Baggaley, J., & Ploetz, T. (Mar, 2019). Towards Reliable, Automated General Movement Assessment for Perinatal Stroke Screening in Infants Using Wearable Accelerometers. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, *3*(1), 1–11. https://doi.org/10.1145/3314399

Garello, L., Moro, M., Tacchino, C., Campone, F., Durand, P., Blanchi, I., … Odone, F. (2021). A Study of At-term and Preterm Infants' Motion Based on Markerless Video Analysis. In *29th European Signal Processing Conference*, 1196–1200. IEEE https://doi.org/10.23919/EUSIPCO54536.2021.9616293

Georgevici, A. I., & Terblanche, M. (Feb, 2019). Neural networks and deep learning: A brief introduction. *Intensive Care Medicine*, *45*(5), 712–714. https://doi.org/10.1007/s00134-019-05537-w

Gima, H., Shimatani, K., Nakano, H., Watanabe, H., & Taga, G. (Jun, 2019). Evaluation of Fidgety Movements of Infants Based on Gestalt Perception Reflects Differences in Limb Movement Trajectory Curvature. *Physical Therapy*, *99*(6), 701–710. https://doi.org/10.1093/ptj/pzz034

Gong, X., Li, X., Ma, L., Tong, W., Shi, F., Hu, M., Zhang, X.-P., … Yang, C. (Dec, 2022). Preterm infant general movements assessment via representation learning. *Displays*, *75*(1), 1–8. https://doi.org/10.1016/j.displa.2022.102308

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Gori, M., Monfardini, G., & Scarselli, F. (2005). A new model for learning in graph domains. In *Proceedings IEEE International Joint Conference on Neural Networks,* 729–734. IEEE https://doi.org/10.1109/IJCNN.2005.1555942

Gravem, D., Singh, M., Chen, C., Rich, J., Vaughan, J., Goldberg, K., … Patterson, D. (Jun, 2012). Assessment of Infant Movement With a Compact Wireless Accelerometer System. *Journal Of Medical Devices, 6*(2), 1–10. https://doi.org/10.1115/1.4006129

Groos, D., Adde, L., Aubert, S., Boswell, L., De Regnier, R.-A., Fjørtoft, T., … Støen, R. (July, 2022). Development and Validation of a Deep Learning Method to Predict Cerebral Palsy from Spontaneous Movements in Infants at High Risk. *JAMA Network Open*, *5*(7), 1–14. https://doi.org/10.1001/jamanetworkopen.2022.21325

Groos, D., Adde, L., Stoen, R., Ramampiaro, H., & Ihlen, E. (Jan, 2022). Towards human-level performance on automatic pose estimation of infant spontaneous movements. *Computerized Medical Imaging And Graphics*, *95*(1), 1–14. https://doi.org/10.1016/j.compmedimag.2021.102012

Haataja, L., Mercuri, E., Regev, R., Cowan, F., Rutherford, M., Dubowitz, V., & Dubowitz, L. (Aug, 1999). Optimality score for the neurologic examination of the infant at 12 and 18 months of age. *The Journal of Pediatrics*, *135*(2), 153–161. https://doi.org/10.1016/S0022-3476(99)70016-8

Hadders-Algra, M. (Aug, 2004). General movements: A window for early identification of children at high risk for developmental disorders. *The Journal of Pediatrics*, *145*(2), S12–18. https://doi.org/10.1016/j.jpeds.2004.05.017

Hadders-Algra, M. (Feb, 2021). Early Diagnostics and Early Intervention in Neurodevelopmental Disorders - Age-Dependent Challenges and Opportunities. *Journal of Clinical Medicine*, *10*(4), 11–23. https://doi.org/10.3390/jcm10040861

Hajian-Tilaki, K. (Jan, 2013). Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation. *Caspian Journal of Internal Medicine*, *4*(2), 627–635. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3755824/

Hashimoto, Y., Furui, A., Shimatani, K., Casadio, M., Moretti, P., Morasso, P., & Tsuji, T. (2022). Automated Classification of General Movements in Infants Using Two-Stream Spatiotemporal Fusion Network. In *Medical Image Computing and Computer Assisted Intervention*, 753–762. Springer. https://doi.org/10.1007/978-3-031-16434-7_72

Hesse, N., Bodensteiner, C., Arens, M., Hofmann, U. G., Weinberger, R., & Sebastia, A. (2019). Computer Vision for Medical Infant Motion Analysis: State of the Art and RGB-D Data Set. In *Computer Vision – ECCV 2018 Workshops,* 32–49. Springer. https://doi.org/10.1007/978-3-030-11024-6_3

Hesse, N., Pujades, S., Black, M. J., Arens, M., Hofmann, U. G., & Schroeder, A. S. (Oct, 2020). Learning and Tracking the 3D Body Shape of Freely Moving Infants from RGB-D sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(10), 2540–2551. https://doi.org/10.1109/TPAMI.2019.2917908

Hesse, N., Pujades, S., Romero, J., Black, M. J., Bodensteiner, C., Arens, M., … Sebastian, A. (2018). Learning an infant body model from RGB-D data for accurate full body motion analysis. In *Medical Image Computing and Computer Assisted Intervention,* 1–10. Springer. https://doi.org/10.1007/978-3-030-00928-1_89

Iglesias, G., Talavera, E., González-Prieto, Á., Mozo, A., & Gómez-Canaval, S. (Mar, 2023). Data Augmentation techniques in time series domain: A survey and taxonomy. *Neural Computing and Applications*, *35*(14), 10123–10145. https://doi.org/10.1007/s00521-023-08459-3

Ihlen, E. A. F., Støen, R., Boswell, L., Regnier, R.-A. de, Fjørtoft, T., Gaebler-Spira, D., … Adde, L. (Dec, 2019). Machine Learning of Infant Spontaneous Movements for the Early Prediction of Cerebral Palsy: A Multi-Site Cohort Study. *Journal of Clinical Medicine*, *9*(1), 1–6. https://doi.org/10.3390/jcm9010005

Janocha, K., & Czarnecki, W. M. (Feb, 2017). On Loss Functions for Deep Neural Networks in Classification. *Schedae Informaticae, 25*(1), 49–59. https://doi.org/0.4467/20838476SI.16.004.6185

Jardine, L., Mausling, R., Caldararo, D., Colditz, P., & Davies, M. (Nov, 2022). Accelerometer measures in extremely preterm and or extremely low birth weight infants and association with abnormal general movements assessments at 28- and 32-weeks postmenstrual age. *Early Human Development*, *174*(1)*,* 1–13. https://doi.org/10.1016/j.earlhumdev.2022.105685

Ji, S., Ma, D., Pan, L., Wang, W., Peng, X., Amos, J. T., … Ren, P. (2023). Automated Prediction of Infant Cognitive Development Risk by Video: A Pilot Study. In *IEEE Journal of Biomedical and Health Informatics*, 1–12. IEEE. https://doi.org/10.1109/JBHI.2023.3266350

K. D. McCay, E. S. L. Ho, C. Marcroft, & N. D. Embleton. (2019). Establishing Pose Based Features Using Histograms for the Detection of Abnormal Infant Movements. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 5469–5472. IEEE. https://doi.org/10.1109/EMBC.2019.8857680

K. D. McCay, E. S. L. Ho, D. Sakkos, W. L. Woo, C. Marcroft, P. Dulson, & N. D. Embleton. (2021). Towards Explainable Abnormal Infant Movements Identification: A Body-part Based Prediction and Visualisation Framework. In *2021 IEEE EMBS International Conference on Biomedical and Health Informatics*, 1–4. IEEE. https://doi.org/10.1109/BHI50953.2021.9508603

K. D. McCay, E. S. L. Ho, H. P. H. Shum, G. Fehringer, C. Marcroft, & N. D. Embleton. (Jan, 2020). Abnormal Infant Movements Classification With Deep Learning on Pose-Based Features. *IEEE Access*, *8*(1), 51582–51592. https://doi.org/10.1109/ACCESS.2020.2980269

K. D. McCay, P. Hu, H. P. H. Shum, W. L. Woo, C. Marcroft, N. D. Embleton, A. Munteanu, & E. S. L. Ho. (Jul, 2022). A Pose-Based Feature Fusion and Classification Framework for the Early Prediction of Cerebral Palsy in Infants. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *30*(1), 8–19. https://doi.org/10.1109/TNSRE.2021.3138185

Kim, S. A., Lee, Y. J., & Lee, Y. G. (Dec, 2011). Predictive Value of Test of Infant Motor Performance for Infants based on Correlation between TIMP and Bayley Scales of Infant Development. In *Annals of Rehabilitation Medicine*, *35*(6), 860–866. https://doi.org/10.5535/arm.2011.35.6.860

Kingma, D. P., & Ba, L. J. (2015). Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations,* 1–10. ICLR. https://doi.org/10.48550/arXiv.1412.6980

Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations,* 10–21. ICLR. https://doi.org/10.48550/arXiv.1609.02907

Kukačka, J., Golkov, V., & Cremers, D. (2017). Regularization for Deep Learning: A Taxonomy. In *International Conference on Learning Representations,* 1–10. ICLR. https://doi.org/10.48550/arXiv.1710.10686

Kwong, A. K. L., Fitzgerald, T. L., Doyle, L. W., Cheong, J. L. Y., & Spittle, A. J. (Feb, 2018). Predictive validity of spontaneous early infant movement for later cerebral palsy: A systematic review. *Developmental Medicine and Child Neurology*, *60*(5), 480–489. https://doi.org/10.1111/dmcn.13697

Li, M., Wei, F., Li, Y., Zhang, S., & Xu, G. (Mar, 2021). Three-Dimensional Pose Estimation of Infants Lying Supine Using Data from a Kinect Sensor with Low Training Cost. *IEEE Sensors Journal*, *21*(5), 6904–6913. https://doi.org/10.1109/JSEN.2020.3037121

Liashchynskyi, P., & Liashchynskyi, P. (Dec, 2019). Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. *arXiv,* 1–24. https://doi.org/10.48550/arXiv.1912.06059

Ludwig, P. E., Reddy, V., & Varacallo, M. (2022). Neuroanatomy, Central Nervous System (CNS). In *StatPearls*. StatPearls Publishing. http://www.ncbi.nlm.nih.gov/books/NBK442010/

Luo, T., Xiao, J., Zhang, C., Chen, S., Tian, Y., Yu, G., Dang, K., & Ding, X. (2022). Weakly Supervised Online Action Detection for Infant General Movements. In *Medical Image Computing and Computer Assisted Intervention*, 721–731. Springer. https://doi.org/10.1007/978-3-031-16434-7_69

Maitre, N. L., Burton, V. J., Duncan, A. F., Iyer, S., Ostrander, B., Winter, S., … Byrne, R. (2020). Network Implementation of Guideline for Early Detection Decreases Age at Cerebral Palsy Diagnosis. *Pediatrics*, *145*(5) 3–19. https://doi.org/10.1542/peds.2019-2126

McCay, K. D., Ho, E. S. L., Marcroft, C., & Embleton, N. D. (2019). Establishing Pose Based Features Using Histograms for the Detection of Abnormal Infant Movements. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* 5469–5472. IEEE. https://doi.org/10.1109/EMBC.2019.8857680

McCay, K., Hu, P., Shum, H., Woo, W., Marcroft, C., Embleton, N., Munteanu, A., & Ho, E. (Mar, 2022). A Pose-Based Feature Fusion and Classification Framework for the Early Prediction of Cerebral Palsy in Infants. *IEEE Transactions On Neural Systems And Rehabilitation Engineering*, *30*(1), 8–19. https://doi.org/10.1109/TNSRE.2021.3138185

Meinecke, L., Breitbach-Faller, N., Bartz, C., Damen, R., Rau, G., & Disselhorst-Klug, C. (Apr, 2006). Movement analysis in the early detection of newborns at risk for developing spasticity due to infantile cerebral palsy. *Human Movement Science*, *25*(2), 125–144. https://doi.org/10.1016/j.humov.2005.09.012

Melo, F. (2013). Area under the ROC Curve. In *Encyclopedia of Systems Biology*. Springer. https://doi.org/10.1007/978-1-4419-9863-7_209

Moro, M., Pastore, V., Tacchino, C., Durand, P., Blanchi, I., Moretti, P., Odone, F., & Casadio, M. (Nov, 2022). A markerless pipeline to analyze spontaneous movements of preterm infants. *Computer Methods And Programs In Biomedicine*, *226*(1), 1–9. https://doi.org/10.1016/j.cmpb.2022.107119

Murtagh, F. (Jul, 1991). Multilayer perceptrons for classification and regression. *Neurocomputing*, *2*(5), 183–197. https://doi.org/10.1016/0925-2312(91)90023-5

Nguyen-Thai, B., Le, V., Morgan, C., Badawi, N., Tran, T., & Venkatesh, S. (Oct, 2021). A Spatio-Temporal Attention-Based Model for Infant Movement Assessment From Videos. *IEEE Journal Of Biomedical And Health Informatics*, *25*(10), 3911–3920. https://doi.org/10.1109/JBHI.2021.3077957

Ni, H., Xue, Y., Ma, L., Zhang, Q., Li, X., & Huang, S. X. (Jan, 2023). Semi-supervised body parsing and pose estimation for enhancing infant general movement assessment. *Medical Image Analysis*, *83*(1), 102654–102658. https://doi.org/10.1016/j.media.2022.102654

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.

Noble, Y., & Boyd, R. (Jan, 2012). Neonatal assessments for the preterm infant up to 4 months corrected age: A systematic review. *Developmental Medicine And Child Neurology*, *54*(2), 129–139. https://doi.org/10.1111/j.1469-8749.2010.03903.x

Novak, I., Morgan, C., Adde, L., Blackman, J., Boyd, R., Brunstrom-Hernandez, J., … Badawi, N. (Sep, 2017). Early, Accurate Diagnosis and Early Intervention in Cerebral Palsy Advances in Diagnosis and Treatment. *JAMA Pediatrics*, *171*(9), 897–907. https://doi.org/10.1001/jamapediatrics.2017.1689

Orlandi, S., Raghuram, K., Smith, C. R., Mansueto, D., Church, P., Shah, V., Luther, M., & Chau, T. (2018). Detection of Atypical and Typical Infant Movements using Computer-based Video Analysis. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* 3598–3601. IEEE. https://doi.org/10.1109/EMBC.2018.8513078

O'Shea, K., & Nash, R. (May, 2015). An Introduction to Convolutional Neural Networks. arXiv, 1–21 https://doi.org/10.48550/arXiv.1511.08458

Page, M. J., Moher, D., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., … McKenzie, J. E. (Mar, 2021). PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews. *BMJ*, *372*(1), 160–168. https://doi.org/10.1136/bmj.n160

Peyton, C., Pascal, A., Boswell, L., deRegnier, R., Fjørtoft, T., Støen, R., & Adde, L. (Oct, 2021). Inter-observer reliability using the General Movement Assessment is influenced by rater experience. *Early Human Development*, *161*(1), 1–12. https://doi.org/10.1016/j.earlhumdev.2021.105436

Philippi, H., Karch, D., Kang, K., Wochner, K., Pietz, J., Dickhaus, H., & Hadders-Algra, M. (May, 2014). Computer-based analysis of general movements reveals stereotypies predicting cerebral palsy. *Developmental Medicine And Child Neurology*, *56*(10), 960–967. https://doi.org/10.1111/dmcn.12477

Pino, M. C., Donne, I. L., Vagnetti, R., Tiberti, S., Valenti, M., & Mazza, M. (Jun, 2022). Using the Griffiths Mental Development Scales to Evaluate a Developmental Profile of Children with Autism Spectrum Disorder and Their Symptomatologic Severity. *Child Psychiatry & Human Development, 55*(1)*,* 117–126. https://doi.org/10.1007/s10578-022-01390-z

Prechtl, H. F. (1977). *The Neurological Examination of the Full-term Newborn Infant*. Mac Keith Press.

Prechtl, H. F. (Sep, 1990). Qualitative changes of spontaneous movements in fetus and preterm infant are a marker of neurological dysfunction. *Early Human Development*, *23*(3), 151–158. https://doi.org/10.1016/0378-3782(90)90011-7

Prechtl, H. F. (Dec, 2001). General movement assessment as a method of developmental neurology: New paradigms and their consequences. *Developmental Medicine and Child Neurology*, *43*(12), 836–842. https://doi.org/10.1017/s0012162201001529

Prechtl, H. F., Einspieler, C., Cioni, G., Bos, A. F., Ferrari, F., & Sontheimer, D. (May, 1997). An early marker for neurological deficits after perinatal brain lesions. *Lancet*, *349*(9062), 1361–1363. https://doi.org/10.1016/S0140-6736(96)10182-3

Prechtl, H. F., & Hopkins, B. (Dec, 1986). Developmental transformations of spontaneous movements in early infancy. *Early Human Development*, *14*(4), 233–238. https://doi.org/10.1016/0378-3782(86)90184-2

Q. Wu, P. Qin, J. Kuang, F. Wei, Z. Li, R. Bian, C. Han, & G. Xu. (Feb, 2023). A Training-Free Infant Spontaneous Movement Assessment Method for Cerebral Palsy Prediction Based on Videos. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *31*(1), 1670–1679. https://doi.org/10.1109/TNSRE.2023.3255639

Raghuram, K., Orlandi, S., Church, P., Luther, M., Kiss, A., & Shah, V. (Jun, 2022). Automated Movement Analysis to Predict Cerebral Palsy in Very Preterm Infants: An Ambispective Cohort Study. *Children*, *9*(6), 843–851. https://doi.org/10.3390/children9060843

Raghuram, K., Orlandi, S., Shah, V., Chau, T., Luther, M., Banihani, R., & Church, P. (Aug, 2019). Automated movement analysis to predict motor impairment in preterm infants: A retrospective study. *Journal of Perinatology*, *39*(10), 1362–1369. https://doi.org/10.1038/s41372-019-0464-0

Rahmati, H., Martens, H., Aamo, O., Stavdahl, O., Stoen, R. & Adde, L. (2015). Frequency-Based Features for Early Cerebral Palsy Prediction. In *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 5187–5190. IEEE. https://ieeexplore.ieee.org/document/7319560

Redd, C. B., Karunanithi, M., Boyd, R. N., & Barber, L. A. (Nov, 2021). Technology-assisted quantification of movement to predict infants at high risk of motor disability: A systematic review. *Research in Developmental Disabilities*, *118*(1), 1–21. https://doi.org/10.1016/j.ridd.2021.104071

Reich, S., Zhang, D., Kulvicius, T., Bölte, S., Nielsen-Saines, K., Pokorny, F. B., ... Marschik, P. B. (May, 2021). Novel AI driven approach to classify infant motor functions. *Scientific Reports*, *11*(1), 9888–9898. https://doi.org/10.1038/s41598-021-89347-5

Romeo, D. M., Ricci, D., Brogna, C., & Mercuri, E. (Aug, 2016). Use of the Hammersmith Infant Neurological Examination in infants with cerebral palsy: A critical review of the literature. *Developmental Medicine and Child Neurology*, *58*(3), 240–245. https://doi.org/10.1111/dmcn.12876

Rosenblatt, F. (1957). The Perceptron: A Perceiving and Recognizing Automaton. (Technical report) Cornell Aeronautical Laboratory.

Ruder, S. (May, 2017). An overview of gradient descent optimization algorithms. *arXiv*, 1–18. https://doi.org/10.48550/arXiv.1609.04747

Saleh, K., Szénási, S., & Vámossy, Z. (Mar, 2023). Generative Adversarial Network for Overcoming Occlusion in Images: A Survey. *Algorithms*, *16*(3), 3–30. https://doi.org/10.3390/a16030175

Sant, N., Hotwani, R., Palaskar, P., Naqvi, W. M., & Arora, S. P. (Jul, 2021). Effectiveness of Early Physiotherapy in an Infant With a High Risk of Developmental Delay. *Cureus*, *13*(7), 16581–16600. https://doi.org/10.7759/cureus.16581

Schmidt, W., Regan, M., Fahey, M., & Paplinski, A. (Jul, 2019). General movement assessment by machine learning: Why is it so difficult? *Journal of Medical Artificial Intelligence*, *2*(8), 1–7. https://doi.org/10.21037/jmai.2019.06.02

Silva, N., Zhang, D., Kulvicius, T., Gail, A., Barreiros, C., Lindstaedt, S., … Marschik, P. B. (Mar, 2021). The future of General Movement Assessment: The role of computer vision and machine learning – A scoping review. *Research in Developmental Disabilities*, *110*(1), 1–19. https://doi.org/10.1016/j.ridd.2021.103854

Sokołów, M., Adde, L., Klimont, L., Pilarska, E., & Einspieler, C. (Dec, 2020). Early intervention and its short-term effect on the temporal organization of fidgety movements. *Early Human Development*, *151*(1), *1*–6. https://doi.org/10.1016/j.earlhumdev.2020.105197

Stahl, A., Schellewald, C., Stavdahl, O., Aamo, O. M., Adde, L., & Kirkerød, H. (Jul, 2012). An optical flow-based method to predict infantile cerebral palsy. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *20*(4), 605–614. https://doi.org/10.1109/TNSRE.2012.2195030

Støen, R., Songstad, N., Silberg, I., Fjortoft, T., Jensenius, A. & Adde, L. (Jul, 2017). Computer-based video analysis identifies infants with absence of fidgety movements. *Pediatric Research*, *82*(4), 665–670. https://doi.org/10.1038/pr.2017.121

Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep High-Resolution Representation Learning for Human Pose Estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 5686–5696. https://doi.org/10.1109/CVPR.2019.00584

Te Velde, A., Tantsis, E., Novak, I., Badawi, N., Berry, J., Golland, P., … Morgan, C. (Aug, 2021). Age of Diagnosis, Fidelity and Acceptability of an Early Diagnosis Clinic for Cerebral Palsy: A Single Site Implementation Study. *Brain Sciences*, *11*(8), 1074–1081. https://doi.org/10.3390/brainsci11081074

Theodoridis, S. (2015). *Machine Learning: A Bayesian and Optimization Perspective*. Academic Press.

Tian, Y., & Zhang, Y. (Apr, 2022). A comprehensive survey on regularization strategies in machine learning. *Information Fusion*, *80*(1), 146–166. https://doi.org/10.1016/j.inffus.2021.11.005

Tong, W., Yang, C., Li, X., Shi, F., & Zhai, G. (2022). Cost-Effective Video-Based Poor Repertoire Detection for Preterm Infant General Movement Analysis. In *5th International Conference on Image and Graphics Processing*, 51–58. Association for Computing Machinery. https://doi.org/10.1145/3512388.3512396

Trevethan, R. (Nov, 2017). Sensitivity, Specificity, and Predictive Values: Foundations, Pliabilities, and Pitfalls in Research and Practice. *Frontiers in Public Health*, *5*(1), 307–318. https://doi.org/10.3389/fpubh.2017.00307

Tsuji, T., Nakashima, S., Hayashi, H., Soh, Z., Furui, A., Shibanoki, T., Shima, K., & Shimatani, K. (Jan, 2020). Markerless Measurement and Evaluation of General Movements in Infants. *Scientific Reports*, *10*(1), 1–15. https://doi.org/10.1038/s41598-020-57580-z

Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (Apr, 2022). A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*, *9*(2), 187–212. https://doi.org/10.1007/s40745-020-00253-5

Warchoł, D., & Oszust, M. (Apr, 2022). Augmentation of Human Action Datasets with Suboptimal Warping and Representative Data Samples. *Sensors*, *22*(8), 2947–2961. https://doi.org/10.3390/s22082947

Ward, I. R., Joyner, J., Lickfold, C., Guo, Y., & Bennamoun, M. (Sep, 2022). A Practical Tutorial on Graph Neural Networks. *ACM Computing Surveys*, *54*(10), 1–35. https://doi.org/10.1145/3503043

Wu, Q., Xu, G., Wei, F., Chen, L., & Zhang, S. (Jan, 2021). RGB-D Videos-Based Early Prediction of Infant Cerebral Palsy via General Movements Complexity. *IEEE Access*, *9*(1), 42314–42324. https://doi.org/10.1109/ACCESS.2021.3066148

Wu, Q., Xu, G., Wei, F., Kuang, J., Zhang, X., Chen, L., & Zhang, S. (Mar, 2021). Automatically Measure the Quality of Infants' Spontaneous Movement via Videos to Predict the Risk of Cerebral Palsy. *IEEE Transactions on Instrumentation and Measurement*, *70*(1), 1–9. https://doi.org/10.1109/TIM.2021.3125980

Wu, Y.-C., van Rijssen, I. M., Buurman, M. T., Dijkstra, L.-J., Hamer, E. G., & Hadders-Algra, M. (Apr, 2021). Temporal and spatial localisation of general movement complexity and variation - Why Gestalt assessment requires experience. *Acta Paediatrica*, *110*(1), 290–300. https://doi.org/10.1111/apa.15300

Xin, C., Kim, S., & Park, K. S. (2023). A Comparison of Machine Learning Models with Data Augmentation Techniques for Skeleton-based Human Action Recognition. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 1–6. Association for Computing Machinery. https://doi.org/10.1145/3584371.3612999

Yan, S., Xiong, Y., & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, 7444–7452. AAAI Press. https://dl.acm.org/doi/10.5555/3504035.3504947

Z. Cao, G. Hidalgo, T. Simon, S. Wei, & Y. Sheikh. (Jan, 2021). OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *43*(1), 172–186. https://doi.org/10.1109/TPAMI.2019.2929257

Zhang, H., Ho, E., & Shum, H. (Dec, 2022). CP-AGCN: Pytorch-based attention informed graph convolutional network for identifying infants at risk of cerebral palsy ? *Software Impacts*, *14*(1), 1–7. https://doi.org/10.1016/j.simpa.2022.100419

Zhang, H., Shum, H. P. H., & Ho, E. S. L. (2022). Cerebral Palsy Prediction with Frequency Attention Informed Graph Convolutional Networks. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, 1619–1625. IEEE. https://doi.org/10.1109/EMBC48229.2022.9871230

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (Nov, 2019). Graph convolutional networks: A comprehensive review. *Computational Social Networks*, *6*(1), 11–38. https://doi.org/10.1186/s40649-019-0069-y

Zhang, X.-M., Liang, L., Liu, L., & Tang, M.-J. (Jul, 2021). Graph Neural Networks and Their Current Applications in Bioinformatics. *Frontiers in Genetics*, *12*(1), 1–9. https://www.frontiersin.org/articles/10.3389/fgene.2021.690049

Zhao, Z., Liu, W., Xu, Y., Chen, X., Luo, W., Jin, L., … Gao, S. (2021). Prior Based Human Completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7947–7957. IEEE. https://doi.org/10.1109/CVPR46437.2021.00786

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., … Sun, M. (Jul, 2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*(1), 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

Zhu, M., Men, Q., Ho, E. S. L., Leung, H., & Shum, H. P. H. (2021). Interpreting Deep Learning based Cerebral Palsy Prediction with Channel Attention. In *IEEE EMBS International Conference on Biomedical and Health Informatics*, 1–4. IEEE. https://doi.org/10.1109/BHI50953.2021.9508619

APPENDIX A

Table 35. Search strings specific to each queried database.

Database	Full search string
Embase	('ai':ab,ti,kw OR 'features':ab,ti,kw OR 'computer-based':ab,ti,kw OR 'video**':ab,ti,kw OR 'sensor*':ab,ti,kw OR 'automat*':ab,ti,kw OR 'acceleromet*':ab,ti,kw OR 'inertial measurement unit':ab,ti,kw OR 'imu':ab,ti,kw OR 'motion analysis':ab,ti,kw OR 'instrumented':ab,ti,kw OR 'deep*learning':ab,ti,kw OR 'machine*learning':ab,ti,kw) AND ('general movement* assessment':ab,ti,kw OR 'gma':ab,ti,kw OR 'spontaneous movement*':ab,ti,kw OR 'fidgety movement*':ab,ti,kw OR 'writhing movement*':ab,ti,kw) AND ('infant*':ab,ti,kw OR 'newborn*':ab,ti,kw OR 'child*':ab,ti,kw OR 'preterm':ab,ti,kw OR 'neonate*':ab,ti,kw OR 'neonatal':ab,ti,kw OR 'cerebral palsy':ab,ti,kw OR 'high-risk':ab,ti,kw) AND [2012-2023]/py
Pubmed	('AI'[Title/Abstract] OR 'features'[Title/Abstract] OR 'computer-based'[Title/Abstract] OR 'video**'[Title/Abstract] OR 'sensor*'[Title/Abstract] OR 'automat*'[Title/Abstract] OR 'acceleromet*'[Title/Abstract] OR 'inertial measurement unit'[Title/Abstract] OR 'imu'[Title/Abstract] OR 'motion analysis'[Title/Abstract] OR 'instrumented'[Title/Abstract] OR 'deep*learning'[Title/Abstract] OR 'machine*learning'[Title/Abstract]) AND ('general movement* assessment'[Title/Abstract] OR 'gma'[Title/Abstract] OR 'spontaneous movement*'[Title/Abstract] OR 'fidgety movement*'[Title/Abstract] OR 'writhing movement*'[Title/Abstract]) AND ('infant*'[Title/Abstract] OR 'newborn*'[Title/Abstract] OR 'child*'[Title/Abstract] OR 'preterm'[Title/Abstract] OR 'neonate*'[Title/Abstract] OR 'neonatal'[Title/Abstract] OR 'cerebral palsy'[Title/Abstract] OR 'high-risk'[Title/Abstract]) AND (2012:2023[pdat])
Scopus	TITLE-ABS-KEY ((_{ai}_OR_{features}_OR _{computer-based}_OR_{video**}_OR_{sensor*}_OR _{automat*}_OR_{acceleromet*}_OR_{inertial measurement unit}_OR_{imu}_OR_{motion analysis}_OR_{instrumented}_OR_{deep*learning}_OR_{machine*learning}_)AND(_{general movement* assessment}_OR_{gma}_OR_{spontaneous movement*}_OR_{fidgety movement*}_OR_{writhing movement*}_) AND (_{infant*}_OR_{newborn*}_ OR _{child*}_ OR _{preterm}_ OR _{neonate*}_ OR _{neonatal}_ OR _{cerebral palsy}_ OR _{high-risk}_)) AND PUBYEAR > _2011_
Web of Science	TS=(('ai' OR 'features' OR 'computer-based' OR 'video**' OR 'sensor*' OR 'automat*' OR 'acceleromet*' OR 'inertial measurement unit' OR 'imu' OR 'motion analysis' OR 'instrumented' OR 'deep*learning' OR 'machine*learning') AND ('general movement* assessment' OR 'gma' OR 'spontaneous movement*' OR 'fidgety movement*' OR 'writhing movement*') AND ('infant*' OR 'newborn*' OR 'child*' OR 'preterm' OR 'neonate*' OR 'neonatal' OR 'cerebral palsy' OR 'high-risk'))
IEEE Xplore	((("All Metadata":"ai" OR "All Metadata":"features" OR "All Metadata":"computer-based" OR "All Metadata":"video*" OR "All Metadata":"sensor*" OR "All Metadata":"automat*" OR "All Metadata":"acceleromet*" OR "All Metadata":"inertial measurement unit" OR "All Metadata":"imu" OR "All Metadata":"motion analysis" OR "All Metadata":"instrumented" OR "All Metadata":"deep*learning" OR "All Metadata":"machine*learning")) AND ("All Metadata":"general movement* assessment" OR "All Metadata":"gma" OR "All Metadata":"spontaneous movement" OR "All Metadata":"fidgety movement" OR "All Metadata":"writhing movement") AND ("All Metadata":"infant" OR "All Metadata":"newborn" OR "All Metadata":"child*" OR "All Metadata":"preterm" OR "All Metadata":"neonate" OR "All Metadata":"neonatal" OR "All Metadata":"cerebral palsy" OR "All Metadata":"high-risk"))
ACM Digital Library	"query": { AllField:(("AI" OR "features" OR "computer-based" OR "video**" OR "sensor*" OR "automat*" OR "acceleromet*" OR "inertial measurement unit" OR "imu" OR "motion analysis" OR "instrumented" OR "deep*learning" OR "machine*learning") AND ("general movement* assessment" OR "gma" OR "spontaneous movement*" OR "fidgety movement*" OR "writhing movement*") AND ("infant*" OR "newborn*" OR "child*" OR "preterm" OR "neonate*" OR "neonatal" OR "cerebral palsy" OR "high-risk")) } "filter": { E-Publication Date: (01/01/2012 TO 12/31/2023) }

Table 36. Subjects, age at birth and monitoring, assessment type, and neurodevelopmental outcome of studies on the writhing period and both (*).
Affiliated studies are grouped in different colors. Shared publicly available datasets are mentioned only by name.

Study	Population of infants	Assessment type and evaluation	Neurodevelop-mental outcome
Gravem et al., 2012	**N:** 10 preterm; **Age(b):** x = 27.1wGA; **Age(m):** x = 36.3wGA	Prechtl; CS(6) / N(4)	None
Doroniewicz et al., 2020	**N:** 36 full-term; **Age(b):** 38-42wGA; **Age(m):** 2nd/3rd day after birth	Prechtl; PR(14) / N(17) (+5 undecided)	None
Fontana et al., 2021	**N:** 68 high-risk; **Age(b):** x = 31.25wGA; **Age(m):** x = 42.1wPMA	Prechtl; CS(8) / PR(17) / N(43)	None
Garello et al., 2021	**N:** 68 (55 preterm, 13 full-term); **Age(b):** n/s; **Age(m):** 40wGA	Prechtl; Ab(27) / N(28) (+13 n/s)	BSID (n/s version) at 2yrs
Moro et al., 2022	**N:** 142 preterm; **Age(b):** x = 29wGA; **Age(m):** 40wGA	Prechtl; Ab (n/s) / N (n/s)	BSID (n/s version at 2yrs + MRI at birth
Hashimoto et al., 2022	**N:** 100 infants; **Age(b):** 30-42wGA; **Age(m):** n/s	Prechtl; PR(27) / N(73)	None
Tong et al., 2022	**N:** 30 newborns; **Age(b):** n/s; **Age(m):** n/s	Prechtl; PR (n/s) / N (n/s)	None
Gong et al., 2022	Pmi-GMA dataset		
Jardine et al., 2022	**N:** 59 EP / ELBW; **Age(b):** x = 26.8wGA; **Age(m):** 28 and 32wPMA	Prechtl; CS, PR, Ch(26) / N(31)	None
Adde et al., 2018**(*)**	**N:** 27 high-risk; **Age(b):** x = 32wGA; **Age(m):** 3-5wPT and 10-15wPT	Prechtl; 3-5wPT: Ab(12) / N(15) 10-15wPT: Ab(0) / N(27)	None
Gao et al., 2019**(*)**	**N:** 34 (21 typically developing (TD), 13 perinatal stroke (PS)); **Age(b):** n/s **Age(m):** monthly from term to 6mCA	Prechtl; TD(21) / PS(13)	None
Tsuji et al., 2020**(*)**	**N:** 19 (3 full-term, 16LBW, 2 n/s); **Age(b):** x = 32wGA **Age(m):** n/s	Prechtl; CS(~5%) / PR(~11%) / N(~82%)	None
Reich et al., 2021**(*)**	**N:** 45 full-term; **Age(b):** x = 39wGA; **Age(m):** 28, 42, 56, 70, 84, 98, and 112 days after birth	Prechtl; FM+(54%) / FM-(46%)	None

Table 37. Subjects, age at birth and monitoring, assessment type, and neurodevelopmental outcome of studies on the fidgety period.
Affiliated studies are grouped in different colors. Shared publicly available datasets are mentioned only by name.

Study	Population of infants	Assessment type and evaluation	Neurodevelopmental outcome
Stahl et al., 2012	**N:** 82 infants; **Age(b):** n/s; **Age(m):** 10-18wPT	N/S; CP(15) / No-CP(67)	N/S at 2 and 5yrs
Adde et al., 2013	**N:** 55 (19 full/near-term, 33 risk-profile); **Age(b):** n/s; **Age(m):** 11wPT and 14wPT	Prechtl; Ab(9) / N(43) (+3 n/s)	European Classification System of CP at 2yrs
Philippi et al., 2014	**N:** 67 (18 low, 49 high-risk); **Age(b):** m = 35wGA; **Age(m):** 3mCA;	Hadders-Algra; CP(10) / No-CP(57)	Bayley Scales of Infant Development-III at 2yrs
Rahmati et al., 2015	**N:** 78 (64 healthy, 14 CP); **Age(b):** n/s **Age(m):** 10-18wPT	N/S; CP(14) / No-CP(64)	N/s method at 2 and 5yrs

Study	Population of infants	Assessment type and evaluation	Neurodevelopmental outcome
Støen et al., 2017	**N:** 150 high-risk; **Age(b):** m = 26.9wGA; **Age(m):** m = 52wPMA	Prechtl; Ab(24.1%) / N(75.9%)	None
Orlandi et al., 2018	**N:** 127 preterm/LBW; **Age(b):** m = 27.7wGA; **Age(m):** 3-5mCA	Hadders-Algra; Ab(29) / N(98)	BSID-III motor composite
Ihlen et al., 2019	**N:** 377 high-risk; **Age(b):** x = 26.3wGA; **Age(m):** m = 12wCA	Prechtl; Ab(31.9%) / N(68.1%)	European Classification System of CP at 2yrs + MRI + cUS
K. D. McCay et al., 2019	MINI-RGBD dataset		
Raghuram et al., 2019	**N:** 152 preterm/LBW; **Age(b):** m = 27.7wGA; **Age(m):** x = 3.81wCA	Hadders-Algra; Ab(32) / N(120)	BSID-III motor composite
Chambers et al., 2020	Chambers dataset		Bayley Infant Neuro-developmental Screener
K. D. McCay et al., 2020	MINI-RGBD dataset		
K. D. McCay et al., 2021	MINI-RGBD dataset		
Nguyen-Thai et al., 2021	**N:** 235 infants; **Age(b):** n/s; **Age(m):** 14-15wPT	Prechtl; Ab(35) / N(200)	None
Wu, Xu, Wei, Kuang, et al., 2021	MINI-RGBD dataset		
D. Sakkos et al., 2021	MINI-RGBD and RVI-25 dataset		
Zhu et al., 2021	MINI-RGBD dataset		
Wu, Xu, Wei, Chen, et al., 2021	**N:** 47 infants; **Age(b):** n/s; **Age(m):** 7-17w(n/s) + MINI-RGBD dataset	Prechtl; Ab(14) / N(33)	None
Groos, Adde, Stoen, et al., 2022	**N:** 557 high-risk; **Age(b):** x = 35.3wGA; **Age(m):** x = 11.8wCA	Prechtl; CP(84) / No-CP(473)	Surveillance of CP in Europe decision tree
Luo et al., 2022	**N:** 757 high-risk; **Age(b):** n/s **Age(m):** x = 55wGA;	Prechtl; FM-(353) / FM+(404)	None
K. D. McCay et al., 2022	MINI-RGBD and RVI-38 dataset		
Raghuram et al., 2022	**N:** 252 preterm/LBW; **Age(b):** m = 27wGA; **Age(m):** 3-5mCA	Prechtl/Hadders-Algra; Ab(41) / N(211)	CP diagnosis via mixed methods
Zhang, Ho, et al., 2022	MINI-RGBD and RVI-38 dataset		
Zhang, Shum, et al., 2022	MINI-RGBD and RVI-38 dataset		
Q. Wu et al., 2023	MINI-RGBD and RVI-38 dataset		
Ji et al., 2023	**N:** 109 preterm; **Age(b):** x = 31.74wGA; **Age(m):** 18.84wCA	Prechtl; Ab(46) / N(48)	BSID-II

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Pesquisa e Pós-Graduação
Av. Ipiranga, 6681 – Prédio 1 – Térreo
Porto Alegre – RS – Brasil
Fone: (51) 3320-3513
E-mail: propesq@pucrs.br
Site: www.pucrs.br