Neural Networks 196 (2026) 108406

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

The recurrent Hopfield mass model: Scalability and optimization

Martina Ferrazza [a,b,c,1,*], Giorgio Gosti [c,d,e,1], Edoardo Milanetti [c,d,f], Sara Spadone [g], Giancarlo Ruocco [c,d,f], Stefania Della Penna [b,**]

[a] International School of Advanced Studies, University of Camerino, Camerino, Italy
[b] DNISC and ITAB, 'G. D'Annunzio' University of Chieti-Pescara, Chieti, Italy
[c] Center for Life Nano- and Neuro-Science, Istituto Italiano di Tecnologia, Rome, Italy
[d] Soft and Living Matter Laboratory, Institute of Nanotechnology, CNR, Rome, Italy
[e] DHI Lab, Istituto di Scienze del Patrimonio Culturale, CNR, Rome, Italy
[f] Department of Physics, Sapienza University of Rome, Rome, Italy
[g] UniCamillus-Saint Camillus International University of Health Sciences, Rome, Italy

ARTICLE INFO

Keywords:
Effective connectivity
Magnetoencephalography
Hopfield network
Hyperparameters search
Recurrent neural networks

ABSTRACT

The Recurrent Hopfield Mass Model (RHoMM) is a novel generative model developed for the estimation of large-scale effective connectivity from magnetoencephalography (MEG) Band Limited Power signals. This binary mass model, compared to the other popular generative models, has the advantages of being data-driven and in principle scalable to encode large-scale systems interactions.

The aim of this work is to evaluate the scalability of the RHoMM and to optimize it at different network sizes (20-200 nodes). Specifically, using simulated objective networks with different architectures, we analysed the effects of L1 rows' normalization and of the addition of a bias in the training. We obtained that in RHoMM without normalization wider intervals of larger learning rates and convergence speeds, associated with lower errors in the inference of the effective connectivity matrix, can be obtained compared to the implementation with normalization, independently from the objective network architecture. Thus, the selection of the learning rate in this case is less critical than with normalization. To also validate the model on experimental data, we employed a dataset of MEG recordings from 10 subjects and a network size of 155 nodes. The obtained results suggest that the model could be effectively scaled to estimate standard-size MEG connectomes.

1. Introduction

Evaluating the brain's anatomical and functional organization and understanding its architecture of communication is one of the main challenges of the neuroscience community. While physical models of single spikes have been proposed and tested for decades (Hodgkin, 1952), the processes that support collective neural behaviour in large-scale cortical systems are still unclear (Breakspear, 2017).

From the perspective of network neuroscience and graph theory (Basset, 2017), the neuronal system can be described by a set of structural and functional modules connected through hubs with specific temporal dynamics (de Pasquale, 2018; de Pasquale, 2012; de Pasquale, 2016; de Pasquale, 2021). It has been shown that spontaneous brain activity is organized in functionally specialized large-scale networks (resting state networks, RSNs) and that this functional organization of the human brain at rest reflects its dynamical capabilities (Deco, 2013), based on the balance of the integration and the segregation of different areas and functions (Friston, 2011a). Unravelling the mechanisms mediating long distance cerebral communication holds the key to deciphering how information is processed, integrated, and transmitted throughout the brain. Yet, its complexity makes this task very challenging.

In this context, generative modelling aims to unravel the mechanisms of brain communication through the generation of synthetic network architectures which produce activity dynamics with the same properties of the collected data, allowing for the mapping from consequences to causes through hidden brain states (Friston, 2011a; Magrans de Abril, 2018). However, the identification of generative models

* Corresponding author.
** Corresponding author at: DNISC and ITAB, 'G. D'Annunzio' University of Chieti-Pescara, via dei Vestini 31, Chieti, 6610, Italy.
E-mail addresses: ferrazza.martina@gmail.com (M. Ferrazza), stefania.dellapenna@unich.it (S. Della Penna).
[1] These two authors equally contributed to the paper

https://doi.org/10.1016/j.neunet.2025.108406
Received 26 May 2025; Received in revised form 2 October 2025; Accepted 28 November 2025
Available online 5 December 2025

providing the underlying patterns of inhibition/excitation (effective connectivity) is still a debated issue. Among the different classes of generative models, Neural Mass Models (NMM) are extremely useful, since they abstract the activity of neural populations into aggregate variables, allowing a simplification of the complex interactions within neural populations while capturing essential features of brain dynamics (Breakspear, 2017).

Different NMMs have been developed to explain the functional organization of Resting State Networks both with fMRI and MEG/EEG (Betzel, 2016; Betzel, 2017). Among the most popular, a variant of Kuramoto model employs coupled and delayed local oscillators to model large-scale connectivity (Cabral, 2011; Cabral, 2014a; Cabral, 2014b). This model though is based on a-priori knowledge of structural connectivity through DTI supported by assumptions on the oscillation frequency to reduce its complexity. Another commonly used class of generative models is dynamic causal modelling (DCM) (Friston, 2014; Friston, 2011a; Friston, 2011b; Kiebel, 2008), that uses Bayesian inference techniques to estimate effective connectivity. This class of models requires prior knowledge about the neurophysiology of brain networks and has an expensive computational cost due to its complexity. For these reasons, it is mainly used in the context of small networks – especially with MEG data – and applied to larger networks (Frässle, 2017; Frässle, 2018; Frässle, 2021) only with fMRI data and with a set of simplifying assumptions.

In general, most of what we know about large-scale integration comes either from structural studies from which aspects of temporal organization are inferred (Cabral, 2011; Cabral, 2014a; Cabral, 2014b), or from fMRI-based functional studies that allow to study this integration at low temporal resolution (Friston, 2014; Frässle, 2021; Frässle, 2018; Frässle, 2017). On this ground, one of the main challenges is to design a large-scale model of brain activity using electrophysiological methods that allow for higher temporal resolution, such as MEG.

In this context, a novel generative model, called the Recurrent Hopfield Mass Model, has been developed to estimate large-scale effective connectivity encoding the temporal dynamics of MEG band-limited power (BLP) time series (Gosti, 2024). RHoMM is a binary mass model whose dynamics is determined by a Recurrent Hopfield Neural Network (Hopfield, 1982), composed of McCulloch-Pitts neurons fully and asymmetrically connected through a set of weights (the effective connectivity (Friston, 2011a)). While Hopfield models – and recurrent neural networks in general – are typically adopted at the microscale level to model small-scale neuronal population (Hopfield, 1982; Folli, 2018; Gosti, 2019; Leonetti, 2020; Grossberg, 2007; Grossberg, 1975; Lansner, 2009), the application of RHoMM to the meso-scale MEG sources implied a different interpretation of the model design. Its application to a network consisting of a relatively small number of neurons (45) promoted it as a suitable model to be used in system neuroscience.

Compared to the popular generative models, RHoMM has the advantage of being data-driven, since it is not based on any biophysical model or prior, but only on a minimal number of assumptions (Gosti et al., 2024). Specifically, these assumptions are: I) the possibility of binarizing the BLP without affecting their non-linear dynamics and the associated functional connectivity (FC) patterns; II) the ability to predict the activity at a certain time step based only on the previous one; III) negligible autapses; IV) asymmetric connections between the neurons; V) the model is trained using Perceptron Learning training procedure with Back Propagation Through Time (BPTT). The fact that it is less constrained than other assumption-based approaches, allows it to intrinsically model the non-linearity of neural dynamics, without, however, increasing its complexity, since there are fewer parameters and hypotheses involved.

These features make RHoMM theoretically scalable to encode larger network sizes – comparable to functional connectomes – and lead to a framework that could be enriched by additional parameters that can improve the model performance.

In this scenario, the objective of this work is to evaluate the scalability of RHoMM using synthetic networks with up to 200 nodes and to optimize the model through the identification of a convergence range of hyperparameter values that allows a trade-off between performance and computational cost for networks with different sizes. We also explored the effectiveness of normalizing the network weights during RHoMM training, as we already applied the weight normalization in (Gosti et al., 2024). Furthermore, we evaluated the effects of adding a parameter by introducing a bias in the training process of the network, since this might improve the speed of learning and the model ability to store complex patterns.

To further validate the method and strengthen the claim that we can extend the findings reached on synthetic data to real MEG connectomes, we then performed an analysis on experimental MEG data. We used a dataset of 10 subjects (26 runs), and we extracted the BLP on a parcellation scheme of 155 nodes belonging to 9 Resting-State Networks, covering the whole cortex.

2. Materials and methods

2.1. The model

The Recurrent Hopfield Mass Model is a binary mass model designed to estimate large-scale effective connectivity from source-level MEG band limited power (BLP) signals, through the optimization of prediction ability on the experimental data (Gosti, 2024). The dynamic evolution of this novel generative model is determined by a recurrent Hopfield Neural Network (Hopfield, 1982), based on binary McCulloch-Pitts neurons with all-to-all asymmetric connections defined through a set of weights. These weights form a weight matrix, i.e. the effective connectivity matrix, that encodes the network's entire dynamics and is estimated through training.

In this model, at each time sample, the state of the network neurons (1-active, 0-rest) is determined by a weighted sum of the inputs, based only on the previous time sample (t-1):

$$s_i(t) = \theta\left[\sum_{j=1}^{n} J_{ij} s_j(t-1)\right] \tag{1}$$

where $s_i(t)$ is the state of the neuron i, n is the number of neurons, $\theta(x)$ is the activation function (Heaviside step function: $\theta(x) = 0$ for $x \leq 0$, and $\theta(x) = 1$ otherwise), and J_{ij} are the elements of the effective matrix indicating the influence of the node j on the node i, where we assume negligible autapses ($J_{ii} = 0$).

To estimate the effective connectivity and predict the evolution in time of the network activity, the model is trained as a feedforward neural network using Perceptron Learning training procedure (Carpenter, 1989) with Backpropagation Through Time (Werbose, 1990). This training method can be derived from a gradient descent (GD) approach which minimizes a cost function that quantifies the incorrect predictions of the network activity state at time t given the activity state at time t-1 (Gosti, 2024).

During the training process, the model is initialized with random parameters and the weights of the effective connectivity matrix (J_{ij}) are gradually adjusted to encode the observed brain dynamics with as few errors as possible. At each step of the BPTT, the estimated $\overline{J_{ij}}$ is perturbed in order to minimize the cost function using all the transitions predictions:

$$\overline{J_{ij}} - \alpha \frac{dE}{d\overline{J_{ij}}} \tag{2}$$

with

$$E = \frac{1}{m} \sum_{t=2}^{m} \sum_{i=1}^{N} \left(s_i(t) - \overline{s_i}(t)\right)^2 \tag{3}$$

where α is the learning rate, E is the cost function (mean square error), m are the time samples in the time series and N is the number of neurons.

The BPTT algorithm is run for a given number of steps, until the cost function reaches a minimum, which may not be zero due to the model assumptions or intrinsic data noise. We expect this minimum to reach the value zero when we use simulated data, since we are using the same model for the simulations and the estimations, further the noise is zero.

The trained matrix of weights represents the effective connectivity, which encodes the temporal dynamics of MEG binary BLP time series and can be used to generate predictions of the experimental data.

Compared to other popular generative models, this approach has the advantages of being simple, since only three hyperparameters are involved in the training: the gradient descent learning rate α, the number of gradient descent steps N_{step} and the number of transitions N_{trans}. Furthermore, RHoMM is based on a minimal number of assumptions linked to specific data features (Gosti, 2024). These characteristics make the model theoretically scalable to encode large-scale systems interactions and suggest its suitability for extensive applications resulting from the investigation of inhibition/excitation dynamics at different temporal and spatial scales.

2.2. Simulated data

To assess the face validity of the model we used binary time series simulated by applying RHoMM equation (Eq. 1) to synthetic connectivity matrices, called objective matrices. We then performed an hyperparameters search (Bergstra, 2012), where we studied RHoMM behaviour under different working conditions (see section 2.2.2 for more details), in order to optimize its prediction performance (error function) for different objective networks and network sizes. We employed synthetic data to examined whether applying the RHoMM framework to the obtained binary sequences we could faithfully recover the ground-truth synthetic matrices. Notably, while we use Eq. 1 to estimate the N states at time t from the N states at the previous time t-1, it is not obvious that, by applying RHoMM to the generated sequences, we obtain a trained matrix overlapping with the objective one. As a matter of facts, we will show how the selection of different hyperparameters impacts on the ability of RHoMM to retrieve the target matrix, so that RHoMM is successful only for specific partitions of the hyperparameter space.

The same simulation procedure was also adopted in Gosti, 2024 on random dense graphs with 45 nodes. We extended this setup, to assess the face validity of the model on larger networks and different network architectures.

With this purpose, we used four different network models representing theoretical communications architectures potentially underlying Resting State Networks: a small-world architecture, a random dense network and two graphs with different modular structures. This allowed us to explore the functioning of the model with some extreme theoretical cases so that we could reasonably assume its correct functioning in the estimation of intermediate and closer-to-reality architectures. For each architecture we created networks with different sizes, ranging from 20 to 200 nodes, to evaluate the scalability of RHoMM. For every combination of network architecture and size we performed the hyperparameters search, to identify a trade-off between performance and computational cost that could be extended to networks with different architectures and different numbers of nodes.

We operated according to a data analysis pipeline divided in three stages (Fanini, 2024). The overall analysis and optimization process is described in Fig. 1. First, we generated the simulated dynamics from the random networks which we call objective networks. Second, we processed the simulated dynamics to infer the estimated effective connectivity. Third, we compared the estimated and the objective connectivity to evaluate the efficacy of RHoMM.

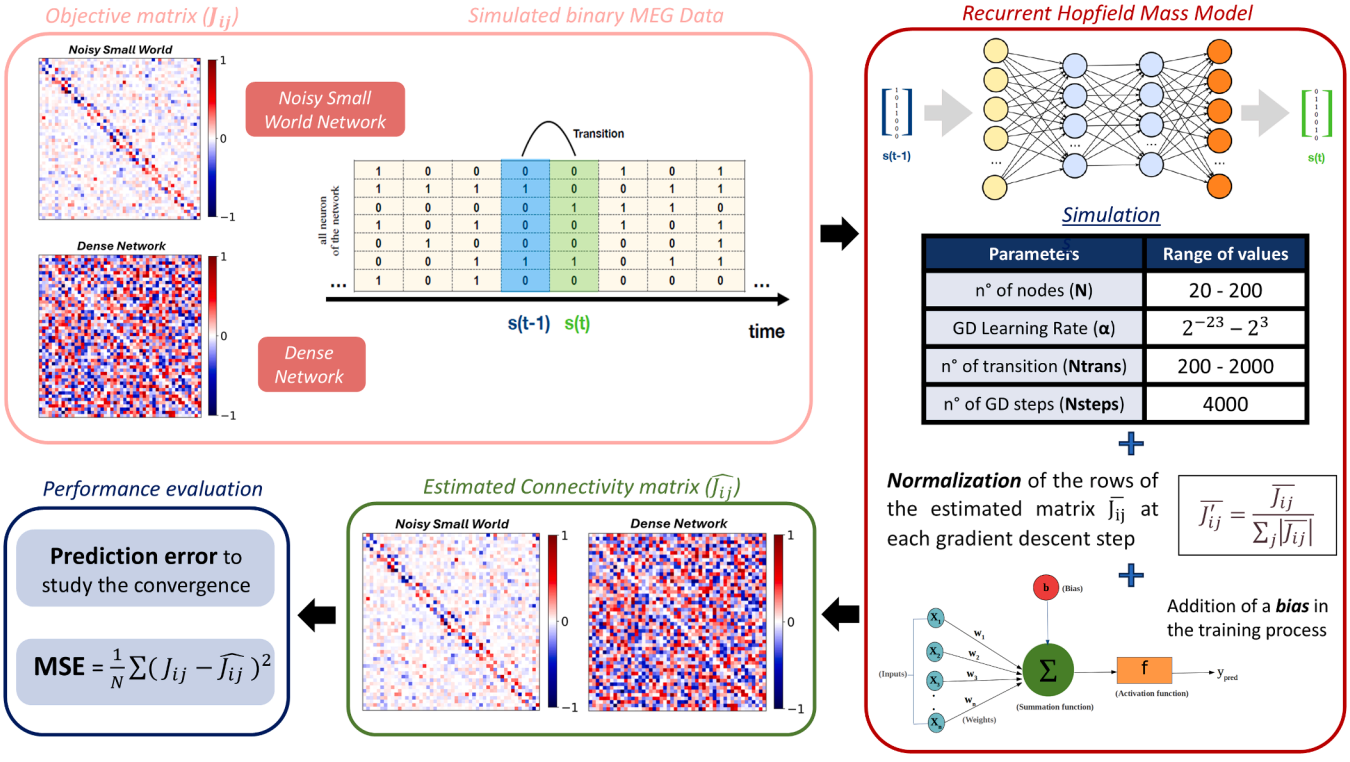

Fig. 1. Overall analysis process. From the objective matrices, we generated simulated binary MEG BLP temporal transitions, that we used as input to the Recurrent Hopfield Mass Model. To evaluate the scalability of the model and its operating range we performed a hyperparameters search for networks with different architectures and different sizes. Moreover, we evaluated the effectiveness of the normalization of the J_{ij} weights at each GD step and we explored the effect of the addition of a bias in the training of the neural network. We computed the prediction error to study the convergence, and we employed the mean square error between the ground truth and estimated matrices as an evaluation metric for the performance.

The simulations and the data analysis were performed using Python 3.10.

2.2.1. Network architectures and MEG simulated data

Following the procedure applied in Gosti et al. (2024), we generated synthetic MEG binary power transition by applying RHoMM equation (Eq. 1) to simulated connectivity matrices with different architectures, to explore RHoMM's ability to retrieve different theoretical connectivity architectures. Specifically, we generated adjacency matrices with four different network models: small-world, dense random, two modules partition and Gaussian modules partition.

Fig. 2 shows the main difference between the architectures: small-orld networks are sparse, have more edges between nearby nodes, and were typically associated with functional and structural connectivity matrices (de Pasquale, 2018, Bullmore and Sporns, 2009). Random networks are used as null models to evaluate statistical significance of topological measures obtained from functional and structural connectivity matrices, and they could also represent a target topology for pathologies where connectivity is altered by dysfunction or degeneration (see e.g. Rubinov, 2009). Two-module networks and Gaussian module networks are modular architectures characterised by different numbers and sizes of modules. They allow the introduction in RHoMM framework of the concept of modularity – a hallmark of real brain functional networks.

As an example, in Fig. 2 we only report the adjacency matrices of the networks with 150 nodes. The complete Fig. showing the other network sizes can be found in the supplementary materials (Fig. S1).

Fig. 2A was made with Aton which is part of the H2IOSC open cloud (Fanini, 2021) and can be interactively viewed with the following URLs: https://aton.ispc.cnr.it/s/ggosti/20240913–UCijcj05o and https://aton.ispc.cnr.it/s/ggosti/20240913-xmxrhg4v6.

The small-world networks were obtained using the Watts-Strogatz algorithm with 4 nearest neighbours and a rewiring probability of 0.3. To make this configuration more realistic, we added some random noise, replacing 30 % of the zero-elements with non-null weights with an amplitude equal to 1/5. This was critical especially in the larger connectivity matrices where having only 4 connected nearest neighbours for each node was far from usual architectures reported in the literature.

For the dense random network, we created model-free matrices with non-zero random weights uniformly distributed between -1 and 1 and then selected a minimum threshold for the weights' amplitude equal to 0.2 and set to zero the values below the threshold.

For the two-modules architecture we used a generalization of the planted-l-partition algorithm described in Fortunato, 2009. We created a partition graph of two modules with N/2 size (where N is the number of nodes of the graph) and with randomly assigned weights, uniformly distributed between -1 and 1. We built the graph imposing a probability of edges of 0.8 within groups and of 0.1 between groups.

The Gaussian random partition graphs were obtained by creating k partitions each with a size drawn from a normal distribution with mean N/6 and variance N/60 (Brandes, 2003). Nodes were connected with the same probability as for the two-modules partition graph, within and between clusters.

From these artificially generated objective networks we generated the synthetic binary MEG BLP temporal transitions applying the RHoMM equation (Eq. 1) to random initial conditions. More specifically, these synthetic binary sequences were generated applying Eq. 1 to T initial conditions of N binary states (1 for active neurons, 0 for resting ones), where N is the number of nodes. These initial conditions were generated randomly, without any constraints on the binary states distribution and without specifying an initial random seed. At each time step, the N binary states s_t^N were generated according to Eq. 1, using the states at the previous time sample (t-1), and each initial random binary state was used to generate the state at the first time sample.

Starting from different random seeds, we generated four replicas of each objective matrix.

These synthetic binary sequences were then the input of RHoMM to retrieve the related estimated Jij matrices.

2.2.2. Optimization of the model hyperparameters

To extensively evaluate RHoMM behaviour when changing the network's size and architecture, we also tested different implementations of the generative model. Specifically, given networks with an increasing number of nodes (from 20 to 200) and following parameter optimization, we evaluated RHoMM ability to encode dynamics of large-scale systems.

For the three hyperparameters involved in the RHoMM training - the gradient descent learning rate, the number of transitions (layers of the network) and the number of gradient descent steps- we adopted the following strategies.

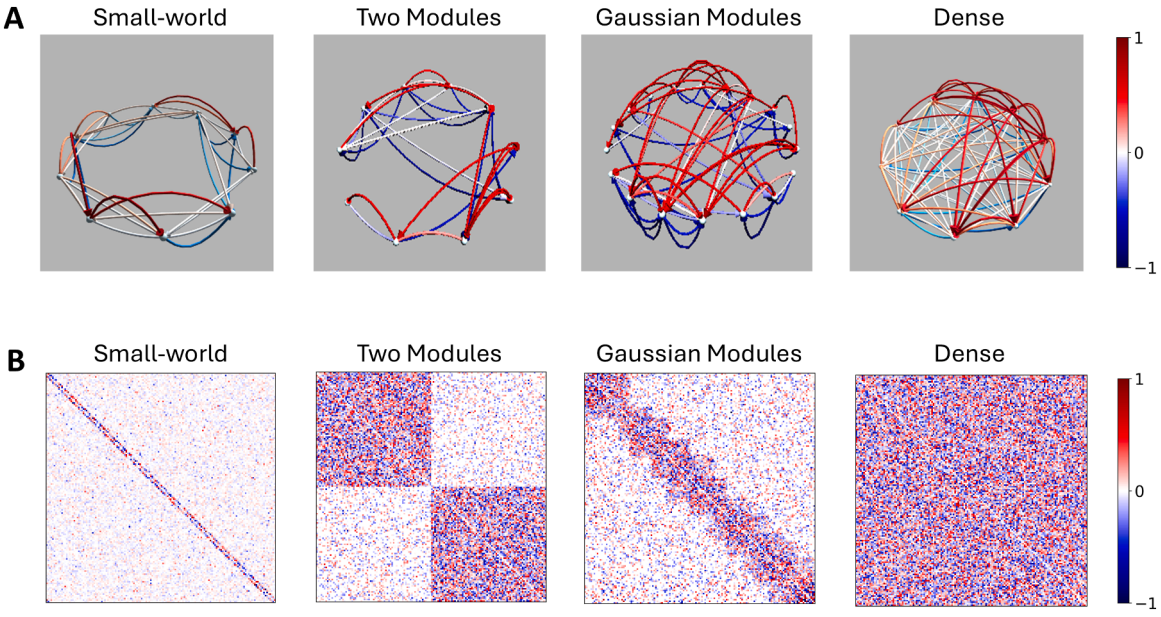

Fig. 2. Network representations. (A) 3D representation of the four different architectures with weights represented with colours. (B) Adjacency matrices with 155 nodes of the four different graph architectures: small-world, two-modules, Gaussian modules and dense random.

For each network architecture with a fixed number of nodes, we tested the algorithm for different values of the learning rate and numbers of temporal transitions (Table 1), to evaluate the changes in the convergence and in the performance of the model. The learning rate α must be chosen carefully because it determines how fast the learning network changes at each iteration of the gradient descent algorithm. More precisely, for each value of learning rate and number of transitions we ran the BPTT for a certain number of steps N_{steps} or until the cost function reached a plateau (Bottou, 2012), that we expect to be zero, since we are working with simulated signals. If the value of α is too small the learning algorithm requires an infeasible time to reach the local optimal value. Contrarily, if the value is too high it can miss the local optima. For larger α the algorithm converges faster to the optimal reconstruction error, but after a critical value of α, it does not converge to the optimal reconstruction anymore. This trend mildly depends on the network architecture.

The maximum number of GD steps was selected close to the optimal values obtained in Gosti et al., 2024, but usually the number of steps needed for the cost function to reach a plateau was smaller than this value and was retained to evaluate RHoMM performance. The search interval of the learning rate and of the numbers of transitions were chosen in a range including the optimal values in the same work. For each hyperparameter, we report the range of values employed in the search in Table 1.

2.2.3. Optimization of RHoMM training

In parallel with the hyperparameters search, we evaluated the impact of the normalization of the J_{ij} weights and the addition of a bias neuron in the model.

Since RHoMM is invariant under the multiplication of each J_{ij} row by an arbitrary positive value a_i (Eq. 4), the original work (Gosti, 2024) includes the L1 normalization of the estimated adjacency matrix at each gradient descent step, using Eq. 5:

$$s_i(t) = \theta\left[\sum_{j=1}^{n} J_{ij}s_j(t-1)\right] = \theta\left[\sum_{j=1}^{n} a_i J_{ij}s_j(t-1)\right] \tag{4}$$

$$\overline{J_{ij}} = \frac{\overline{J_{ij}}}{\sum_j |\overline{J_{ij}}|} \tag{5}$$

The weight normalization should have the aim of preventing an excessive growth of the network weights and of allowing the use of larger learning rates, thus reducing the computational costs (Yuan & Xiao, 2019; Zhang et al., 2015; Liu et al., 2022; Salimans & Kingma, 2016).

However, learning dynamics in neural networks are very sensitive to the norms of the weights and the different weight normalization techniques are still a debated issue and not yet widely adopted (Hoffer et al., 2019; Li et al., 2020; Biswas, 2024).

For this reason, in order to carry out a more exhaustive analysis of the model behaviour for larger networks, we ran the simulations with and without normalization and we compared the results to select the best procedure to adopt for the RHoMM training.

We eventually explored the impact of the addition of a bias in RHoMM training. To investigate if this approach could influence the model behaviour in terms of convergence and performance, we ran the simulation both with and without the bias addition and compared the outcomes.

Table 1
Range of values involved in the hyperparameters search.

Hyperparameters	Ranges of values
GD learning rate (α)	2^{-23} - 2^{3}
GD maximum n° steps (N_{steps})	$4{\cdot}10^{3}$
n° transitions (N_{trans})	200 - 2000

The Eq. 6 describes the evolution of the brain state $s_i(t)$ in each neuron with the addition of the bias b.

$$s_i(t) = \theta[u_i + b_i] \text{ with } u_i = \sum_{j=1}^{n} J_{ij}s_j(t-1) \tag{6}$$

The use of the bias b_i has the effect of applying an *affine transformation* to the output u_i of the linear combiner in the model of a single McCulloch-Pitts neuron (Haykin, 2009). In particular, depending on whether the bias is positive or negative, the argument of the activation function θ is modified as described in Eq. 6. As a result of this transformation, the bias introduces a shift in the activation function that no longer passes through the origin. This means that, in this model, the neuron i would need a higher or lower input intensity to switch to the active state ($s_i(t) = 1$).

The bias b_i is an external parameter of the neuron i but it can be equivalently considered as a new network weight of an added input signal fixed at +1. Training the bias together with the other network weights means training the neurons' activation threshold based on their previous states.

The addition of the bias introduces a new parameter in the model that represents the susceptibility of the network neurons (which in the RHoMM are meso-scale brain areas). This could lead to an improvement in the training outcome, in the speed of learning and in the ability of the model to store complex patterns, as the model could more easily adapt to data. On the other hand, adding a bias introduces additional parameters, increasing the risk of overfitting if the model is already close to the required representation capacity.

2.2.4. Model evaluation and statistical analysis

To study the convergence of the model we analysed the trend of the prediction error (Eq. 3), which is equivalent to the average number of incorrectly predicted items at each time sample, since $s_i(t)$ and $\overline{s_i}(t)$ are binary. In fact, given that we conducted the analysis in a controlled environment using simulated data, we expect the prediction error to reach the zero value in case of convergence. Therefore, for each network we can identify a hyperparameters range of convergence inside which the prediction error is zero.

As an evaluation metric for the performance, we employed the Mean Square Error (MSE) between the ground-truth objective matrix and the estimated one:

$$MSE = \frac{1}{N}\sum \left(J_{ij} - \widehat{J}_{ij} \right)^2 \tag{7}$$

We repeated each simulation using 4 replicas of each matrix architecture and size.

For the small-world networks and the dense matrices, we compared the results obtained using four different model implementations, also testing every combination of hyperparameters:

- without bias and without normalization,
- without bias and with normalization,
- with bias and without normalization,
- with bias and with normalization.

We implemented a Kruskal-Wallis test (Kruskal, 1952) to evaluate the significance of the changes in the convergence and in the performance inside the convergence range caused by the implementation of the normalization procedure and by the addition of the bias. With this purpose we selected the largest value of the learning rate at which the model converged and its related MSE and we compared these values between the different implementations.

For the modular architectures, we evaluated RHoMM convergence and performance only in the setting without normalization and bias, as the simulations with small-world and dense networks had already identified this training configuration as the optimal one. As a matter of

fact, given that small-world and random dense networks represent contrasting extremes with respect to structural properties such as modularity, it is reasonable to assume that the performance trends observed in these two cases can also be generalized to modular architectures.

On the results obtained using the optimal training configuration, we then performed a Mantel test (Mantel, 1967) to determine the significance of the correlation between the objective networks and the estimated ones, establishing in this way the ability of the model to reproduce the data.

2.3. Experimental Data

As described in the previous sections, the objective of using synthetic data was to verify the ability of the model to retrieve a target effective connectivity matrix and to analyse how different hyperparameters could affect this ability, with a ground-truth to compare the results with and a better control and understanding of data characteristics and variability. Although using synthetic data simplifies development, it also risks overestimating model performance on experimental data. As a matter of facts, real MEG data can introduce damaging effects – such as noise, artifacts, volume conduction and nonstationarity – that can lead to distorted results or instability. On the other hand, an experimental dataset could also have beneficial effects (physiological constraints, variability, complexity) that can improve robustness and ecological validity of connectivity estimation.

To further corroborate the hypothesis that the model could reliably estimate the effective connectivity matrix underlying cerebral binarized power time courses and generate binary time series with the same functional properties of the experimental ones, we decided to include in the manuscript also an analysis and model validation on MEG experimental data.

2.3.1. Preprocessing and binarization

We selected a subset of 10 subject from the dataset published in de Pasquale et al., 2021. Each subject underwent two or three resting-state runs (lasting 3 or 5 min), during which the subjects maintained fixation on a small visual target. In total, we analysed 26 runs. Neuro-magnetic signals (filter settings 0.16–250 Hz, 1025 Hz sampling rate) were recorded using the 153-magnetometer MEG system installed in a 4-layer magnetically shielded room at the University of Chieti (Della Penna et al., 2000). The recordings have been preprocessed to separate environmental and physiological (e.g., cardiac, ocular) artifacts from the sensor space MEG time series using an ICA-based approach (Mantini et al., 2011) and retain only ICs that were not artifacts for further analysis. Then, the sensor weights of the non-artifactual ICs maps were projected into the source space (a 3D isotropic Cartesian grid with 4 mm step) using a Weighted Minimum Norm Least Square (WMNLS) estimation. The source vector activity of every voxel in the 3D grid was obtained as the linear combination of IC time courses weighted by the corresponding source maps, consisting of a current dipole with 3 orthogonal directions for each voxel. The individual 3D grid was then projected into the MNI 152 atlas space so that every voxel centroid was assigned to a set of MNI coordinates.

Finally, for a selected set of voxels (see below) source-space Band Limited Power (BLP) time series in the alpha [7-14 Hz] and beta [14–25 Hz] bands were estimated as the sum over the 3 directions of the squared signal averaged over 200 ms windows sliding every 20 ms to cover the entire resting-state run.

The BLP was extracted on a parcellation scheme of 155 nodes belonging to 9 Resting-State Networks, covering the whole cortex (de Pasquale et al., 2021), as shown in Fig. S2 of the supplementary materials: Dorsal and Ventral Attention (D/VAN), Somatomotor (SMN), Visual (VIS), Auditory (AUD), Language (LAN), Default Mode (DMN), Fronto Parietal (FPN), Control (CON).

The fundamental unit of RHoMM is the McCulloch-Pitts neuron, which can only assume two discrete values: zero at rest and one when active. For this reason, to be able to associate each seed voxel with the network neurons, we transformed the BLP continuous signals into binary time series, assigning the value of 1 to significantly large power values as in the following. After normalizing the signal of each seed voxel between 0 and 1, we computed the binarized time course at a fixed threshold, that we imposed equal to two standard deviations above the mean of the envelope distribution. The binarized seed BLP time courses constitute an n-dimensional vector (t), representing a sequence of brain's states, used to estimate the effective connectivity matrix.

To support the scalability hypothesis, we decided to also train the model on a connectome with a sparser parcellation scheme, so that we could directly compare the results obtained with different network sizes. Therefore, we selected a subset of 45 nodes from the 155-nodes parcellation, choosing the nodes closest (based on the Euclidean distance) to the parcelling used in Gosti et al., 2024.

2.3.2. RHoMM validation on experimental data

We estimated the effective connectivity matrices using the following hyperparameters, based on the results obtained with the simulations: Nstep= 1000, Ntrans=700, α= 4, for 45 and 155 nodes. Then, to assess if the effective connectivity matrix estimated by RHoMM captures the collective dynamics of the node set, we produced simulated binary BLP timecourses as in Gosti et al., 2024. Specifically, for each run, we applied the corresponding RoHMM to an initial random state, consisting of 155 binary elements. The initial state was selected randomly from the set of states of the binarized experimental BLP, to have a realistic pattern of active and inactive neurons. This initial state evolved until it reached an attractor producing a binary time sequence with variable length. This procedure was repeated 5000 times, and the resulting time sequences were concatenated to produce a simulated run. Finally, for each band, we compared the correlation matrix estimated from the binarized experimental signals with the one obtained from simulated runs using RHoMM. Specifically, we estimated the Pearson's correlation matrix for each run of both experimental and simulated signals and we normalized the correlation values by applying a z-Fisher transformation.

To evaluate the statistical significance of the comparison between simulated and experimental connectivity matrices, we applied the Mantel test on the average functional connectivity matrices.

3. Results

3.1. Simulated data

3.1.1. Effects of normalization

We first analysed the impact of normalization on the model convergence, without bias. With this purpose, we computed the prediction error at each step of the gradient descent. When the model converges to the optimal reconstruction, the prediction error reaches the value zero, while if the error decreases until a plateau different from zero two phenomena may have occurred: either the learning algorithm needs more steps to reach convergence, or it is missing the local minimum. The outcome depends on the chosen hyperparameters, and it can be influenced by the network size and architecture, and by the normalization process adopted in the training.

Thus, for each network size, we analysed the prediction error trends when varying the gradient descent learning rate and the number of transitions. We applied the model on the dense and the small world network architectures, we ran the simulations both with and without normalization, and we compared the results.

Here, to better understand how the convergence changes for the different implementations, we report the best error, i.e. the value of the prediction error at the last step of the gradient descent. Given that we considered a number of GD steps sufficiently large for the algorithm to reach convergence, if the best error is zero, the model converges while if it is a positive value it means the model reached a plateau different from

zero at the end of the training.

Fig. 3 shows the mean and standard error of the best error computed on the replicas for the dense architecture, without (Fig. 3a) and with normalization (Fig. 3b). For each network size, the best error is represented as a function of the learning rate alpha and the different number of transitions (different colours). From these results we can observe in general that, in the small alpha range, the best error decreases as the learning rate increases until it reaches the value zero when the model converges at the optimal learning rate values. However, while without the normalization the model still converges for a wide range of learning rates up to large values, the normalization has the effect of narrowing the alpha range of convergence. So, even if the prediction error reaches the zero value for smaller learning rate with the normalization, the model stops converging for larger values of alpha. Furthermore, while without normalization the convergence alpha range is only mildly dependent on the number of nodes and transitions, the latter parameters are more critical when the normalization is applied, requiring their fine tuning and a careful selection of alpha especially when the number of nodes is changed. For the dense architecture, in the supplementary materials (Figs. S3 and S4) we show the prediction error trends of each replica, for different learning rates, number of nodes and number of transitions. While the trends without normalization are monotonically converging as alpha increases, with only a tiny dependence on the number of nodes, with normalization the prediction errors are never converging for larger alpha, in an interval that depends on the number of nodes and transitions.

For a smoother narrative, we report the results obtained on the best error for the small-world network in the supplementary materials (Figs. S5a and S5b), since there are no significant differences in the trends.

Notably, since the convergence speed depends on the learning rate and the number of steps required by the algorithm to converge, in Fig. 4 (and Fig. S6) we report the convergence ratio (alpha/convergence steps) as a function of alpha. When no normalization is applied, the convergence ratio is tiny for small alpha and increases rapidly for alpha $>10^{-3}$. Conversely, with normalization this ratio is at most some orders of magnitude lower than with normalization, and in a small alpha interval. Overall these results suggest a faster convergence of the model without normalization.

Furthermore, to evaluate the ability of the model to estimate the objective matrix in the convergence interval and better characterize this interval, we computed the Mean Square Error between the objective and estimated networks. For each network size, we analysed the MSE when varying the gradient descent learning rate and the number of transitions, and we compared the results obtained with and without normalization. In Fig. 5 we can observe the mean and the standard error of the MSE computed on the replicas for the dense architecture, with and without normalization (the results for the small-world network are shown in Fig. S7). The MSE generally decreases while approaching convergence. However, while in the absence of normalization the error stabilizes at the minimum value for increasing learning rates, when the normalization is applied, we can observe a minimum and then an increase in the

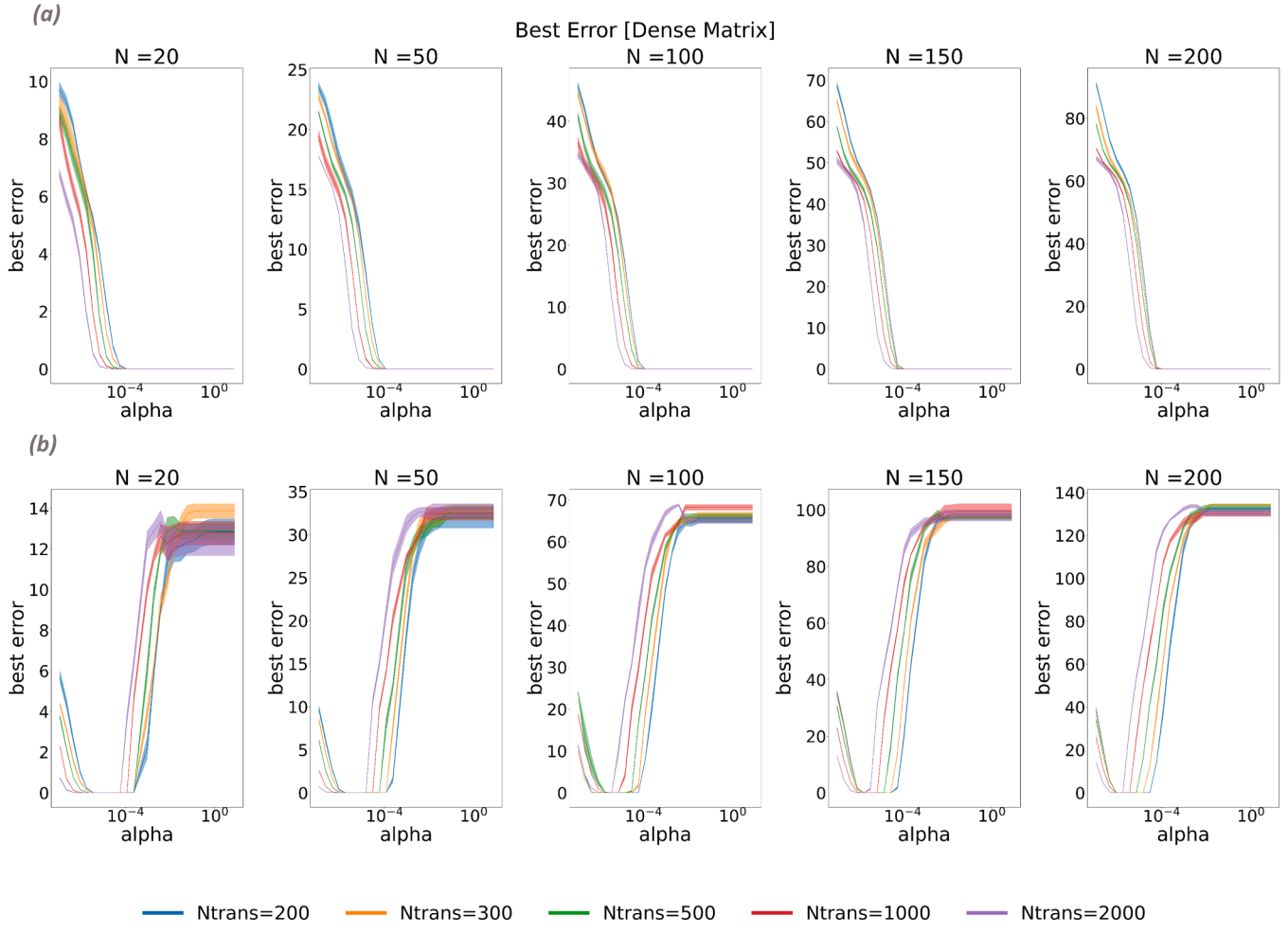

Fig. 3. Best error without bias (Dense network). (a) best error trends for the model without normalization and (b) best error trends obtained for the model with normalization, in the case of the dense architecture. The metric is shown for different network sizes as a function of the GD learning rate (alpha) and of the number of transitions, that are represented by the different colours. The best error is computed as the prediction error at the last step of the training.

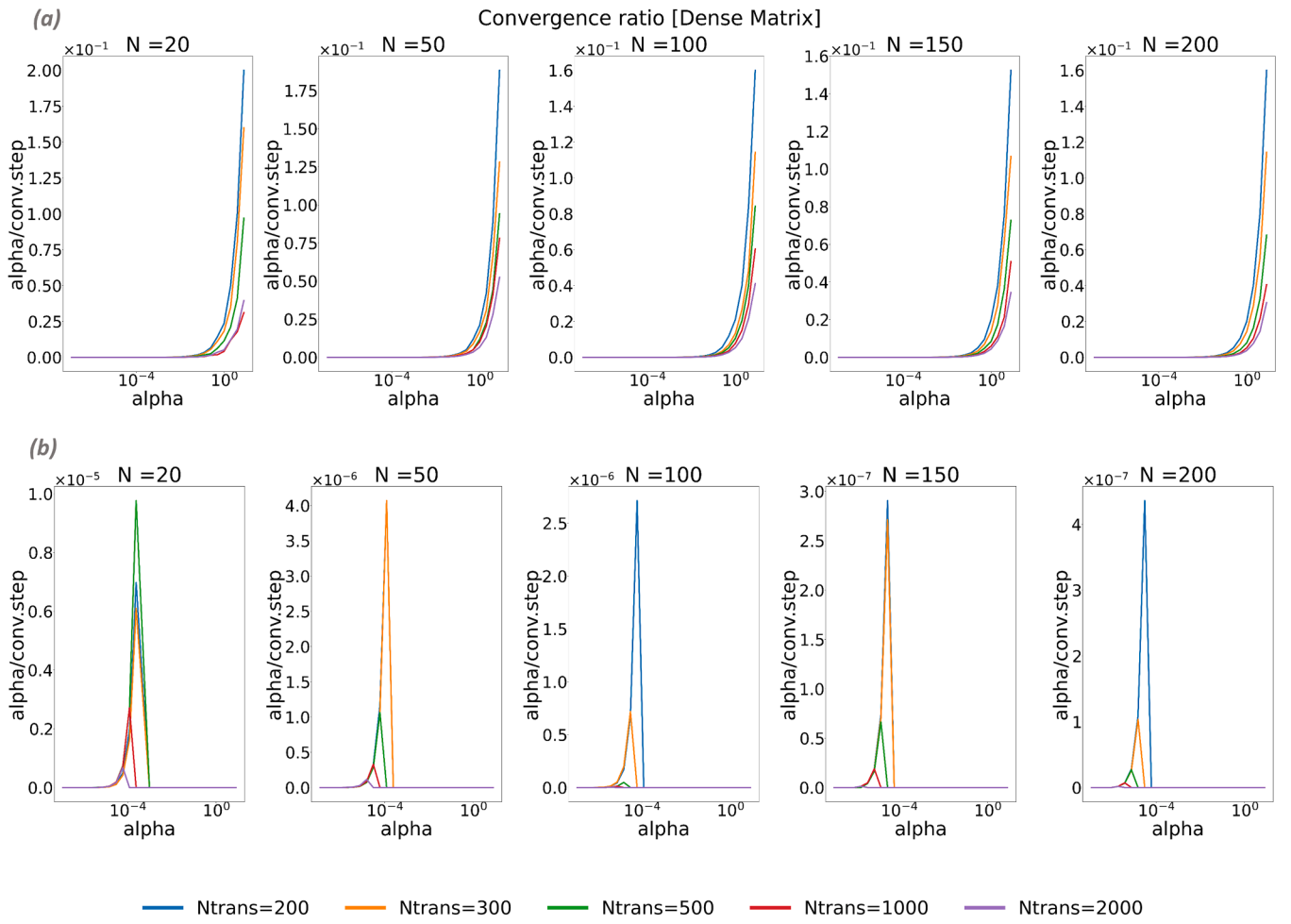

Fig. 4. Convergence ratio (Dense network). Convergence ratio as a function of the learning rate alpha. This parameter is linked to the convergence speed, and it is computed as the ratio between the learning rate alpha and the convergence step (i.e. the number of steps the model needs to reach prediction error = 0). The higher this ratio, the higher the speed of convergence for each alpha. The plots show: (a) convergence ratio for the model without bias and without normalization; (b) convergence ratio for the model without bias and with normalization.

MSE for larger learning rates above the convergence interval. In both cases, the smaller is the number of transitions, the larger is the error, suggesting to preferably select a larger number of transitions. Similar results are obtained for the small-world network (Fig. S7).

Finally, in Fig. 6, we summarize the above results including statistical comparisons among the different implementations of the model. In Fig. 6*[left]* we report how the alpha convergence range for each network size and for the different numbers of transitions varies across the different implementations of the model, for the dense network architecture. From this plot we can better appreciate both the shift of the alpha interval of convergence to lower learning rate values, and the general narrowing of this range when normalization is applied. The latter is confirmed by the Kruskal-Wallis test of the largest values of alpha for which the model converges (Fig. S8*[left]*), suggesting that the model without the normalization converges for significantly higher values of the learning rate than the normalized version (p-value $< 10^{-8}$).

A further plus towards the removal of normalization is obtained when comparing the MSE values inside the convergence interval, shown in Fig. 6*[right]* for the dense network (see Fig. S14 for the small-world). Here, we report for each network size the distribution over transitions for the different model configurations (Fig. 6*[right]*), while in Fig. S8 *[right]* we show the trend of the MSE values obtained at the largest converging alpha as a function of the number of transitions and the different model configurations. Following a Kruskal-Wallis test comparing the distributions obtained without and with normalization, we can observe that for a network size > 20 – both for the dense and the small-world architectures – the exclusion of the normalization from the training leads to a significant decrease in the MSE (p-value < 0.05). In summary, these results suggest that in RHoMM without normalization larger learning rates and convergence speeds compared to the implementation with normalization can be obtained, with also lower errors in the inference of the effective connectivity matrix.

3.1.2. *Effects of the bias addition*

The addition of the bias introduces a new parameter in the model, which represents the susceptibility of the network neurons (i.e. the meso-scale brain areas in the RHoMM). We initially hypothesized that the addition of this information could lead to an improvement in the model, increasing its flexibility and efficacy.

To test this hypothesis, we repeated all the simulations with the addition of the bias in the training of the network. Using the simulated data generated from both network architectures – dense and small-world –, we trained the model again implementing the bias addition, with and without normalization, and for all the different combinations of hyperparameters.

The results did not show significant changes in the convergence and in the performance of the model. A more in-depth analysis is shown in the supplementary materials (Figs. S9 – S12), where it's possible to observe the trends of the best error and the MSE obtained after the addition of the bias for both network architectures. Here, to provide a

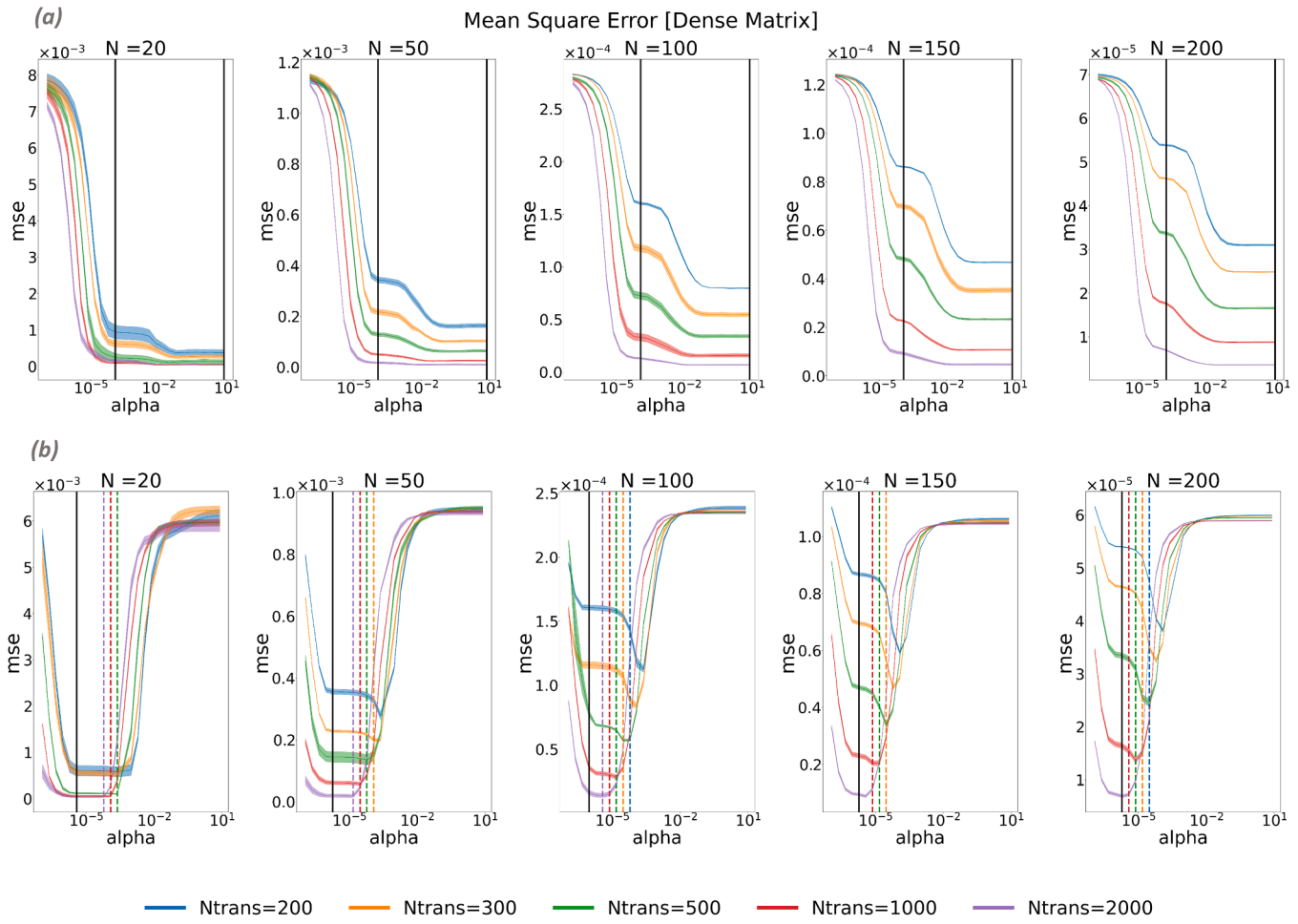

Fig. 5. Mean square error without bias (Dense network). (a) mean square error trends for the model without normalization and (b) mean square error trends obtained for the model with normalization, in the case of the dense architecture. The MSE is shown for different network sizes as a function of the GD learning rate (alpha) and of the number of transitions, that are represented by the different colours. The vertical lines in the plots delimit the alpha convergence interval, i.e. the value of the learning rate where the best error is zero: in (a) the convergence interval is the same for the different numbers of transitions and it is delimited by the black lines; in (b) the alpha convergence interval changes for different numbers of transitions, so we delimited this intervals with different colours indicating the different transitions.

more concise picture, we only show the general trend of the alpha convergence interval and of the MSE at the largest converging alpha compared to the model results without bias (Fig. 6), for the dense network. The same results obtained for the small-world architecture are reported in the Figs. S13 and S14 of the supplementary materials. From the plots we can observe that there are no statistically significant changes in the convergence interval and in the MSE trends between the results obtained with and without bias. As a matter of fact, the Kruskal-Wallis test returned a p-value greater than 0.8 when comparing the model with and without bias both in the normalized and non-normalized case. Furthermore, the test turned out to be significant when comparing the normalized and non-normalized model after the addition of the bias.

Even though the addition of a bias did not lead to an improvement in the model outcome in terms of convergence or ability to reproduce the data, it also did not result in a worsening of the convergence trends (neither in terms of times nor interval width), nor did it lead to a decrease of the MSEs. As a consequence of this we can conclude that the addition of a bias, without negatively impacting the model outcome, still allows to enrich its information content by taking into account the susceptibility of the brain areas modelled by the network neurons.

It is important to point out that we tested the model with the bias addition on objective matrices that were originally built without the bias neuron. So the fact that the error does not increase with an increase in the number of parameters means that it is not overfitting the data, rather it is correctly reproducing the initial objective matrices without the bias neuron.

3.1.3. *Modularity's effects on RHoMM performance and convergence*

The two modular architectures (two-modules and Gaussian modules partition graphs) allow the introduction in RHoMM framework of the concept of modularity – a hallmark of real brain functional networks.

To evaluate the optimal model implementation (i.e. with or without normalization and bias) we only performed the analysis previously described using the small-world and the random dense architecture.

For the modular architectures, we evaluated RHoMM convergence and performance only in the setting without normalization and bias, as the simulations with small-world and dense networks had already identified this training configuration as the optimal one. As a matter of fact, given that small-world and random dense networks represent contrasting extremes with respect to structural properties such as modularity, it is reasonable to assume that the performance trends observed in these two cases can also be generalized to modular architectures.

After training the model without normalization and bias, we analysed the results to verify if there was a consistency with the results obtained with the small-world and random dense objective matrices. With this purpose we computed the convergence ratios, the prediction errors and the MSEs, to evaluate the convergence and the performance

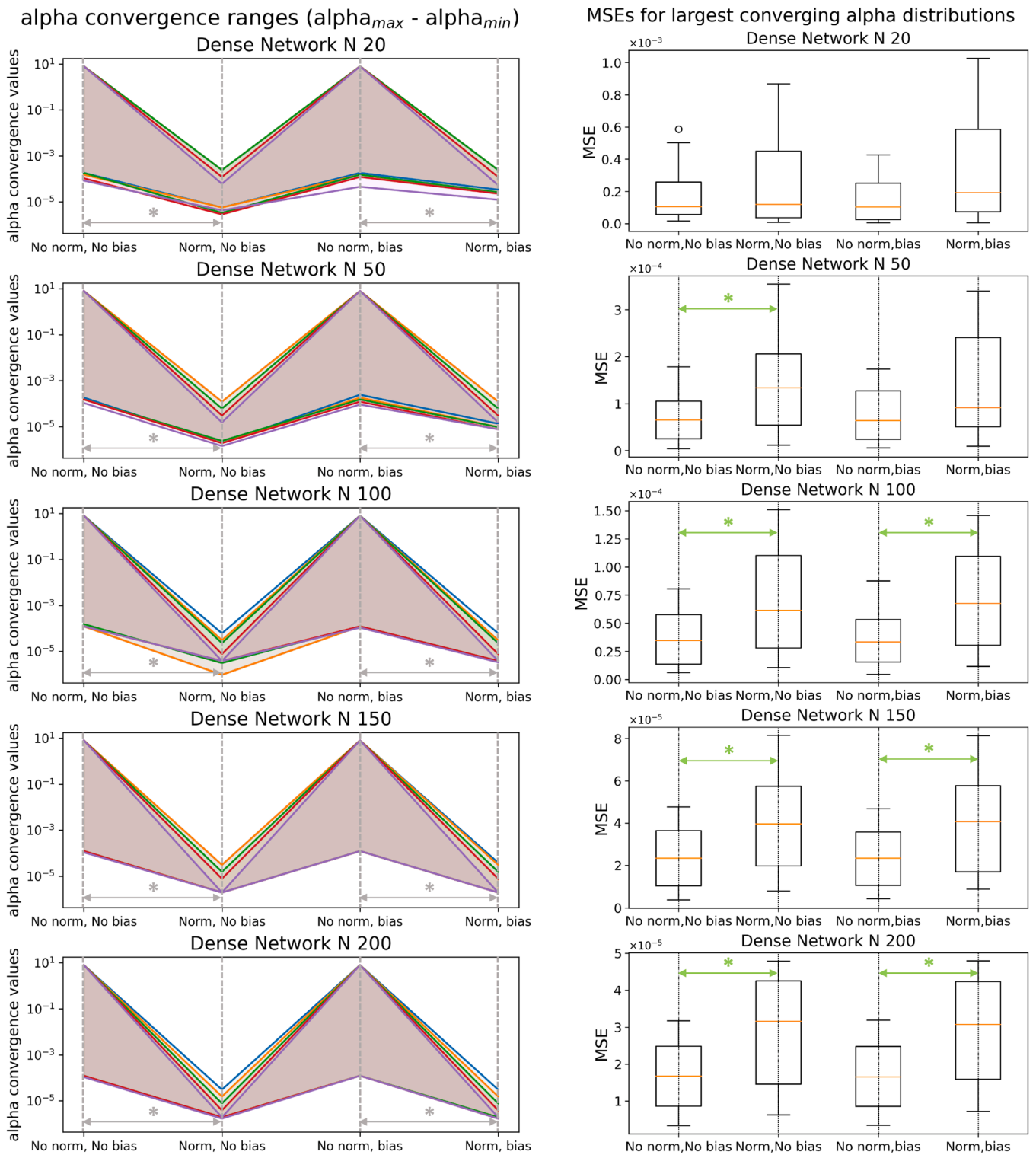

Fig. 6. Convergence and performance of different implementations (Dense network). [left] alpha convergence range changing across different model implementations, for different network sizes, in the case of the dense architecture. The coloured curves represent the maximum and minimum values of convergence for different numbers of transitions. The grey arrows represent the comparisons that resulted significantly different after the Kruskal-Wallis test (* p-value $< 10^{-8}$). [right] for each model implementation, the distributions of the MSEs at the largest converging alpha values, for different network sizes (different plots). The green arrows represent the comparison that results significantly different after the Kruskal-Wallis test (* p-value < 0.05).

of RHoMM for the different sets of nodes, as the hyperparameters change.

We report the results in the supplementary materials (see Figs. S15-S17). From these results we observe similar trends of convergence and performance that we found for the small world and dense architectures, allowing to extend the conclusions reached with these networks to architectures with a clear modular structure.

3.1.4. *Inference of the effective connectivity matrices*

To analyse the ability of the model to reproduce the data over

different network sizes and architectures, we employed a Mantel test, comparing the objective matrices and the ones estimated by the model.

From the results obtained with the small-world and dense architectures, shown in the previous paragraphs, we could see that the exclusion of the normalization process from the training resulted in a widening of the hyperparameters range of convergence (especially the learning rate) and in a significant reduction of the MSEs inside the convergence interval. We could also verify that the addition of the bias in the training did not actually change this trend, nor did it have an impact on the overall convergence and performance. For these reasons, we performed the test on the results obtained for all four network architectures, using the training setting without normalization and without the addition of a bias.

For the dense architecture, in Fig. 7 we show the Mantel test results as a function of the learning rate, for a subset of network sizes and numbers of transitions (see Fig. S18 for the complete results). Specifically, we report the trends of the Pearson's correlation coefficient and the p-value obtained for the different replicas. The test always turned

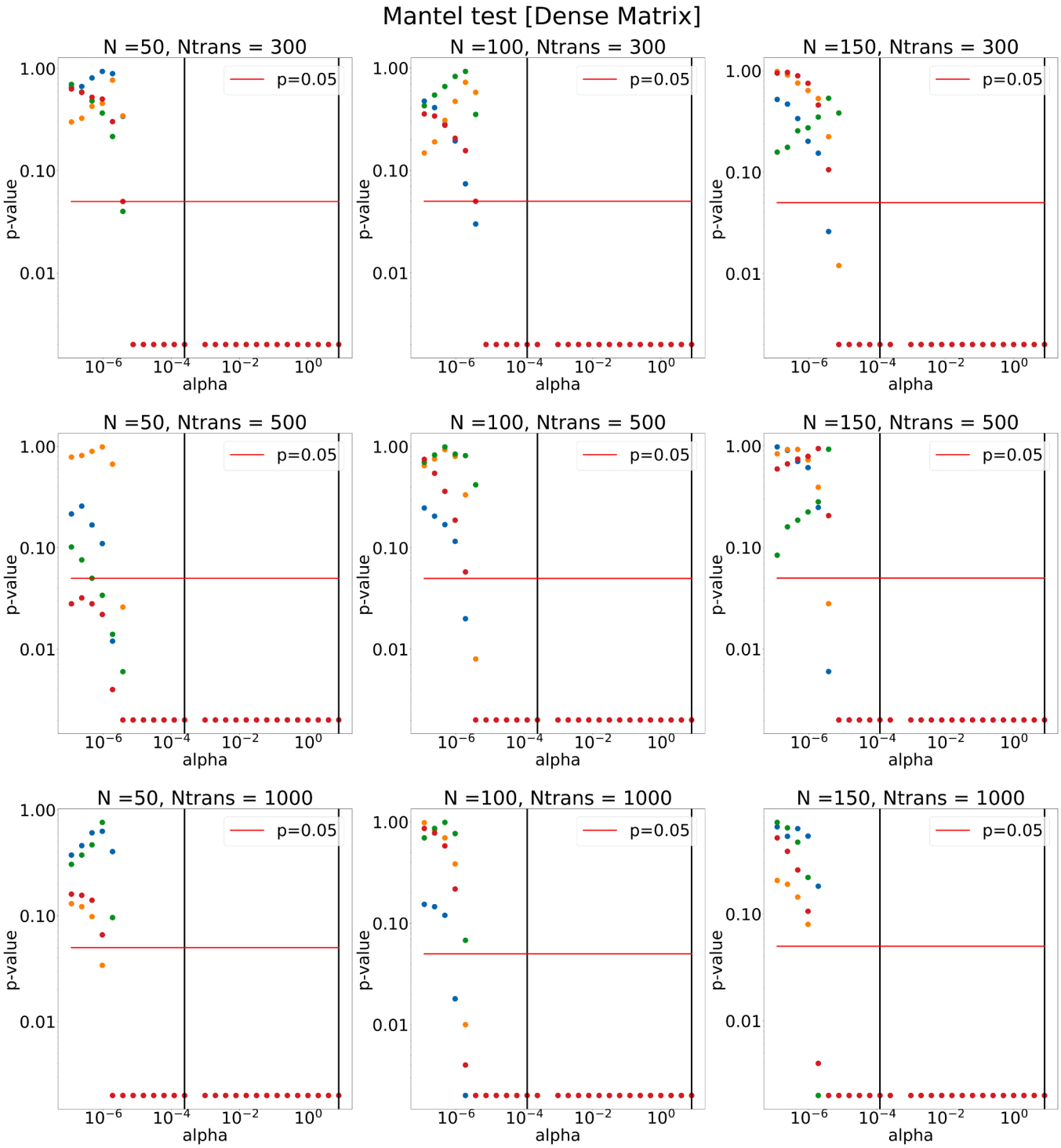

Fig. 7. Mantel test results for the dense architecture, without bias and without normalization. Each plot shows the Pearson's correlation values (purple) and the relative p-values (orange) obtained comparing the different replicas of the objectives and the estimated matrices. The values are represented as a function of the learning rate alpha, for different combinations of number of nodes (N= 50, 100, 200) and number of transitions (Ntrans=300, 500, 1000).

out significant inside the alpha convergence range, for every combination of network size and number of transitions, and the correlation coefficient increases approaching the convergence interval, where it reaches values close to one. A similar trend can be observed for the two-modules partition graph (see Fig. S20 of the supplementary materials). For the small-world network (Fig. S19), we can observe the same trend for numbers of transitions lower than 500, while for shorter sequences the model is not always able to reproduce the data well also inside the alpha convergence range, especially for larger networks. Being the small-world network more diluted than the dense, this behaviour could be linked to a reduced representation capacity of the former, which would need longer input sequences for the training to better reproduce larger networks. The same behaviour is observed in the case of the Gaussian modules partition graph (Fig. S21), where for numbers of transition lower than 500 and for small networks (N<100), the test not always turned out significant inside the convergence interval.

Overall, these results show that, inside the hyperparameters convergence interval, the model is successfully able to reproduce the data over different architectures and network sizes. As an example, in Fig. S22 we report, for each architecture, the comparison between the 100-nodes objective network and the corresponding matrix estimated by RHoMM, obtained with a learning rate of 4 and a number of transitions equal to 1000.

3.2. Experimental data

To further validate the method and strengthen the claim that we can extend the findings reached on synthetic data to real MEG connectomes, we performed an analysis on experimental data. The objective was to corroborate the hypothesis that the model could reliably estimate the effective connectivity matrix underlying cerebral binarized power time courses and generate binary time series with the same functional properties of the experimental ones. With this purpose, we included an analysis and model validation on MEG data obtained from 10 subjects (26 resting-state runs in total), with a parcellation scheme of 155-nodes. To support the validity of the scalability hypothesis on experimental data, we also trained the model on a subset of 45-nodes, selected based on the parcellation scheme used in Gosti et al., 2024 (see Materials and Methods).

We estimated the effective connectivity matrices using the following hyperparameters, based on the results obtained with the simulations: Nstep= 1000, Ntrans=700, α= 4, for 45 and 155 nodes. As we can observe from the Fig. 8A-D and S23A-S23D, the mean over runs of the individual effective connectivity matrices for 155 nodes confirms the inhibitory role of the VIS and DAN networks for both network sizes, consistently with the results shown in Gosti et al., 2024 on the 45-node network.

Then, to assess if the effective connectivity matrix estimated by RHoMM captures the collective dynamics of the node set, we produced simulated binary BLP time courses as in Gosti et al., 2024 (see Materials and Methods for a detailed description). We then compared the correlation matrix estimated from the binarized experimental signals with the one obtained from simulated runs using RHoMM. Figs. 8 and S23 show the mean effective connectome for both parcellations and frequency bands and the related simulated and experimental mean functional connectomes.

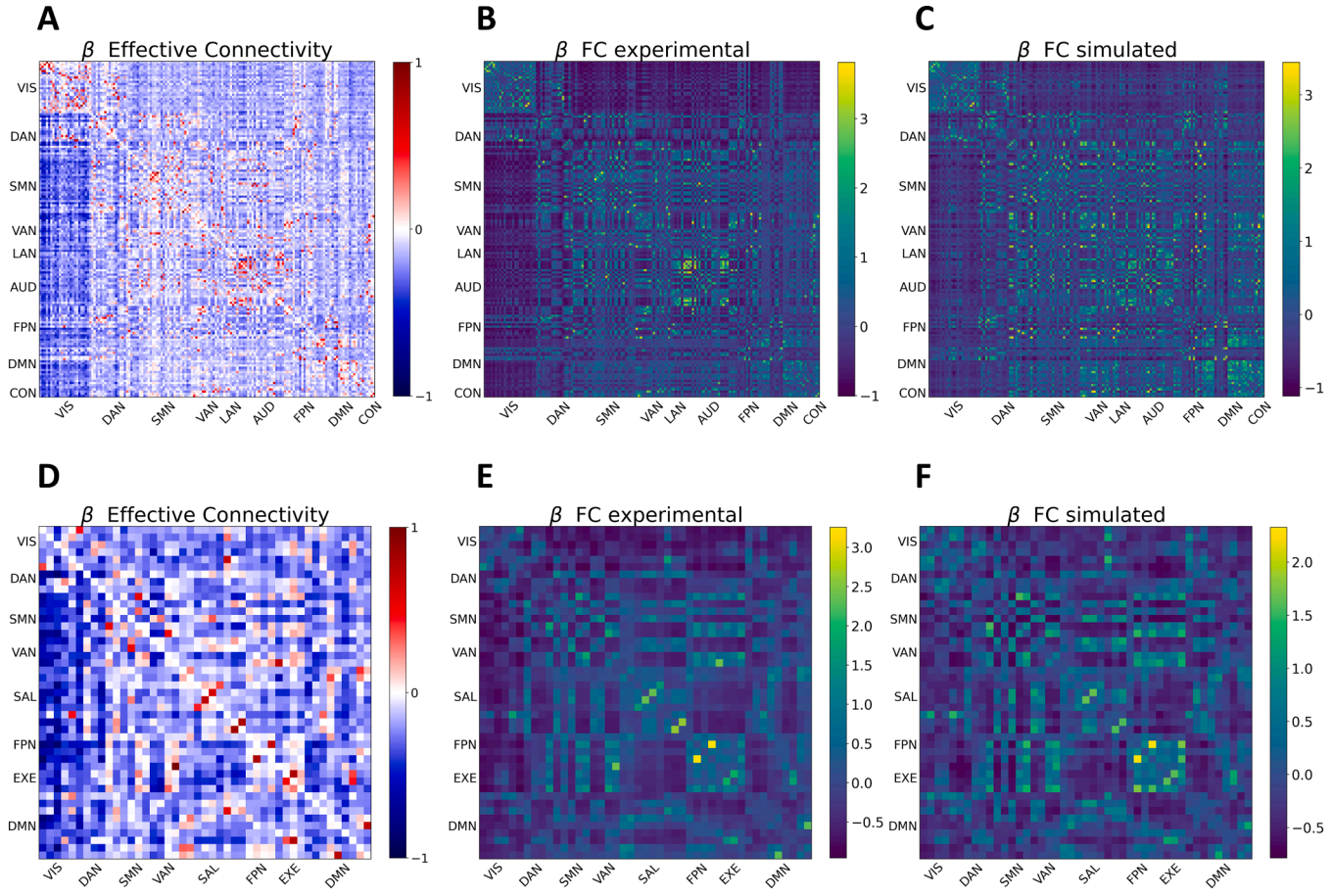

Fig. 8. Comparison between experimental and simulated functional connectivity matrices in the beta band. The effective connectivity matrix estimated with RHoMM for the 155-nodes parcellation (A) and the 45-nodes parcellation (D). The experimental FC matrices for the 155-nodes parcellation (B) and 45-nodes parcellation (E). The simulated FC matrices for the 155-nodes parcellation (C) and 45-nodes parcellation (F).

To evaluate the statistical significance of the comparison between simulated and experimental connectivity matrices, we applied the Mantel test on the average functional connectivity matrices.

In Table 2 we report the Pearson's correlation coefficient obtained from the comparison between experimental and simulated functional connectomes, for the different combinations of network sizes and frequency bands. From the results we can observe that for both parcellations, we obtained a significant similarity through Pearson correlation, both in alpha and beta band ($p < 0.001$). This drives us to the conclusion that the estimated effective connectivity can lead to the generation of signals with reciprocal statistical dependencies similar to those obtained from real data. This evidence, already pointed out in Gosti et al, 2024, can now be extended to real effective MEG connectomes with denser parcellation (comparable with functional connectomes).

4. Discussion

4.1. Aspects of brain dynamics captured by RHoMM

Frameworks that model brain activity as transitions between states are highly effective for characterizing neural dynamics. These approaches generally conceive of "states" as discrete or quasi-stable configurations that capture recurring spatio-temporal patterns of activity. For example, Hidden Markov Modeling (HMM) describes brain activity as switching between latent states defined by characteristic activity or connectivity patterns (Baker et al., 2014; Vidaurre et al., 2017). Dynamic Connectivity Analysis (DCA) emphasizes time-varying coupling between brain regions, often identifying recurrent connectivity patterns as emergent states (Allen et al., 2014; Hutchison et al., 2013). Energy Landscape Analysis (ELA), in turn, represents brain activity as movement within a high-dimensional energy landscape, where local minima correspond to attractor states and probabilistic transitions occur when the system crosses between basins (Watanabe et al., 2014). Together, these methods provide complementary descriptions of how brain networks organize and reconFig. over time.

In these frameworks, a "state" typically refers to a network-level spatio-temporal pattern (e.g., a whole-brain connectivity configuration). In contrast, the RHoMM framework defines "states" as binary vectors describing the activation of all neurons in the network (0 – rest, 1 – active). The system evolves step by step through sequences of these binary vectors, with some neurons switching between active and inactive according to inputs from other neurons and the learned effective connectivity. This connectivity encodes the mechanism that drives transitions from the state at time t to that at t+1. When observing the model's behaviour over time—such as which attractors it settles into or which binary patterns repeat—one begins to see parallels with the "brain states" described in other approaches.

In RHoMM, attractors manifest as quasi-stable binary activation patterns of band-limited power (BLP) across meso-scale nodes (MEG-derived sources). Because the model is a recurrent Hopfield-type system trained on binarized BLP with asymmetric effective connections, its landscape supports both fixed-point–like patterns and limit cycles, rather than only strict fixed points. This phenomenology resonates with evidence that the resting brain explores a landscape of stable and metastable configurations, dwelling in one for some time before transitioning to another (Deco et al., 2019). Accordingly, RHoMM reproduces resting-state dynamics characterized by quasi-stable patterns or cycles with non-trivial dwell times, rather than static attractors alone (Chen et al., 2025). Related models using Hopfield-like dynamics at the macroscale explicitly recover connectome-constrained attractors and data-matched transitions, further supporting this alignment (Englert et al., 2024).

On this basis, a future direction is to examine how the effective connectivity estimated by RHoMM can be used to predict the functional organization of the resting brain over time, linking attractor states in the model with metastable states observed in dynamic functional connectivity.

Table 2
Pearson's correlation values obtained with the Mantel test. All the tests returned a statistically significant outcome with a p-value < 0.001.

Nodes Bands	155	45
Alpha	0.68	0.89
Beta	0.84	0.93

4.2. Simulated data

4.2.1. Scalability to larger networks

An aspect to consider when we analyse generative approaches for the estimation of effective connectivity is the scalability to large network sizes. The simulations and the analysis that we performed showed that the RHoMM reproduces the data well over a wide range of network architectures and sizes, suggesting that RHoMM could represent a valuable tool for the exploration of different aspects of brain's communication with respect to other popular generative models.

As a matter of fact, RHoMM is a generative model with a dynamic evolution based on a recurrent Hopfield neural network, but, while Hopfield networks, and in general recurrent neural networks (RNN), were typically adopted at a microscale level to model small scale neuronal populations (Hopfield, 1982; Amit, 1985; Gosti, 2019; Leonetti, 2020, Grossberg, 2007; Grossberg, 1975; Lansner, 2009), RHoMM is designed to predict large-scale dynamics from MEG BLP. The application of the Hopfield model at different spatial scales in fact required a different interpretation of the Hopfield model and some modifications, such as the application of BPTT and perceptron learning, and the consideration of asymmetric connections, to generate a model capable of reproducing the whole brain activity dynamics (Gosti, 2024).

One of the most popular generative approaches is the dynamic causal modelling (Kiebel, 2008; Friston, 2014), that is based on linear dynamic equations to model the behaviour of neural regions under different conditions. But as a matter of fact, this class of models, was used on MEG data only in the context of small networks (Kiebel, 2008; David, 2006; Lu, 2012; Jafarian, 2020).

In the last years, both RNN-based models (Hahn, 2019; Singh, 2020) and other generative approaches were applied to larger-scale networks in some studies but mainly using fMRI data. For example, Hahn et al. (2019) developed a non-predictive model that reproduces functional connectivity using DTI and a gradient descent approach. One year later, Singh et al. (2020) applied a RNN inspired by a neural mass model (MINDy) to meso-scale fMRI data and were able to estimate the effective connectivity and reproduce brain meso-scale dynamics. In addition, a modified version of DCM, called regression DCM, was developed to analyse larger networks by introducing a convolutional kernel on fMRI BOLD signals to functionally relate it to neural activity (Frässle, 2017; Frässle 2018; Frässle, 2021). As a control analysis in Gosti et al., 2024, we applied an autoregressive dynamical model using the Langevin form of the DCM evolution equation to retrieve the EC matrix from the source space BLP of 45 nodes and generate simulated evolutions. The FC matrix retrieved from the evolutions simulated by RHoMM matched the experimental FC matrix better than the autoregressive dynamical model, that maybe was already critical at that network size. Although these results promoted RHoMM for the analysis of larger MEG BLP networks, a more extensive comparison increasing the network size is worth investigating in the future.

Among the neural mass models there is a variant of Kuramoto model with coupled and delayed local oscillators, that successfully predicted meso-scale activity of both fMRI and MEG/EEG and that, like RHoMM, is scalable to encode larger dynamics. Specifically, this model was able to reproduce fMRI RSN patterns (Cabral, 2011), the spectral content observed by MEG connectivity (Cabral, 2014a), and the slow and

structured amplitude envelopes of band-pass filtered MEG spontaneous signals for networks with 90 nodes (Cabral, 2014b). This model, however, is based on the a-priori knowledge of structural connectivity estimated through diffusion tensor imaging (DTI) and on simplifying assumptions to reduce the complexity. Similarly, the other discussed approaches either rely on the adoption of a specific biophysical model or priors (Cabral, 2014b; Frässle, 2017; Friston, 2011a), either are assumption-based, such as DCM, MINDy or the model described by Hahn et al. (2019).

RHoMM, on the other hand, is designed to model and predict MEG BLP activity though a data-driven and low computational cost approach, that - without constraints on topology, directionality and sparseness - intrinsically models the non-linearity of neural dynamics. These features, together with its simplicity (only three hyperparameters are involved in the training), open to the possibility of scaling the model to larger numbers of neurons (Gosti, 2024), for networks with sizes comparable to functional connectomes ($N > 100$), as we demonstrated in this work with simulated networks. Notably, the MSE decreases with the increasing of the network size (Fig. S8*[right]*), supporting the hypothesis that RHoMM is scalable to encode both small-scale and large-scale systems interactions.

4.2.2. *Effect of weight normalization*

Weight normalization techniques are increasingly employed in deep learning as a form of regularization, to speed up convergence and potentially improve performance (Li et al., 2020; Salimans & Kingma, 2016). These optimization strategies have been developed as an alternative or integration to the most widely adopted Batch Normalization (Ioffe & Szegedy, 2015), whose key assumption of independence between samples falls short in domains with strong correlation between samples, such as recurrent networks.

On this ground, the original RHoMM implementation developed in Gosti, 2024 included the L1 normalization of the estimated adjacency matrix according to Eq. 5.

However, learning dynamics in neural networks are very sensitive to the norms of the weights and it's important to analyse the connection between normalization practices and learning-rate adjustments (Hoffer et al., 2019). As a matter of fact, even if scaling the weights by a positive value does not change the model's equation (see Eq. 4), it does influence the updates that are performed during the training, since it entails a similar scaling in the gradient. In this context, fine tuning the initial learning rate value could help regularise the learning dynamics, mimicking the effect of weight decay (Hoffer et al., 2019; Li et al., 2020; Laarhoven, 2017).

For this reason, in the current work, we decided to further investigate the effects of the normalization in relation to different values of the learning rate, and the network depth and width.

From the results obtained we could observe how the rescaling of weights during the training causes the model to diverge for larger learning rates, while in the absence of normalization, the model keeps converging until the upper limit of the tested learning rates range (Fig. 3).

This behaviour may be linked to instabilities introduced by the rescaling of the weights during the training which, in conjunction with excessively high learning rates, cause a degradation of the model's performance. As a matter of fact, L1 normalization can distort the gradient, which means that the update of the weights is no longer simply a step along the gradient but is influenced by the normalisation itself. This may have non-linear effects that depend on the size of the learning rate and with learning rates that are too high, this distortion can become significant and lead to divergence.

First, since we are not working with integer numbers as the network parameters, the weights length can easily cause a floating-point overflow and lead to a failed training (Li et al., 2020).

Furthermore, the normalization causes a narrowing in the weights space, reducing the flexibility of the model and making learning more difficult, especially with high learning rates (Biswas, 2024). As a matter of fact, large learning rates cause aggressive weight updates, potentially making individual weight differences very large in a single step. Since the renormalization causes the weight updates to be highly dependent on each other, over multiple iterations with too large learning rates this could lead the weight to converge toward uniform values.

Fig. 9 shows an example of a comparison between the trained matrices obtained from the respective objective matrices by training the RHoMM with and without weight normalization, and with a large value of the learning rate (alpha=8), for both the dense (Fig. 9A) and the small-world (Fig. 9B) architectures. In support of the above theory, we can observe that when we employ normalization, for larger learning rates, such as α=8 (outside the alpha converge interval), the weights tend to reach similar values in the same row, reducing the diversity of connections between neurons and potentially degrading the learning capacity of the model.

Without normalization, on the other hand, weights update independently. Large updates, such as those caused by alpha=8, may therefore cause oscillations or instability, but they do not force weights to equalize.

It is important, in conclusion, to underline that row-wise weight normalisation, as well as other weight normalization approaches, is a useful technique to keep weights well scaled, improving model stability and convergence. In fact, if we observe the convergence ratio in Fig. 4, we can see that normalization actually speeds up the convergence for smaller learning rates, as it should.

However, it can have side effects in the presence of high learning rates, as it can reduce the diversity of weights within a row, leading to undesirable behaviour. Hence, when employing weight normalization, it is important to fine tune the learning rate.

4.2.3. *Effects of the bias addition*

Starting from simulated data generated with zero bias, we trained the model using two settings: one without considering a bias term (i.e. forcing the bias = 0) and one with the addition of a bias term that we optimize in the training procedure together with the network weights. We chose this procedure because the simulated data were initially generated with RHoMM with zero bias (i.e. the neuron's threshold were zero). Therefore, if we had attempted to fit zero-bias simulated data with a network that has non-zero bias parameters, the network would have probably learnt weights that compensate for any non-zero biases and that do not match the true generating weights. This is possible because gradient descent tries to minimize the loss and the Hopfield networks have redundancy between weights and biases, i.e. the same fixed point can be realized with different combinations of weights and biases.

When we train the network (via Back Propagation Through Time and gradient descent) including the bias parameters, the loss function no longer depends solely on the weights but also the biases (Liao et al., 2019). Hence if the true generating process had zero bias, then the optimal solution (global minimum of the loss) should correspond to biases of zero and weights that capture the interaction terms. However, due to optimization imperfections, small biases may appear in the trained network even if the true bias is zero, especially if the network is overparametrized and multiple configurations of weights and biases produce the same output (Montazeri & Schmidt, 2024).

To test whether the learned bias is meaningful (improves fit) or incidental, we compared the performance (i.e. the MSE) obtained from the two different training settings described above (see Fig. 6[right]). These results showed no significant differences in the error distributions between the training setting with fixed zero biases and the one with optimized biases. This means that the bias term is not improving generalization, and it can be treated as noise. However, it is also important to point out that the parameter estimation error does not significantly increase with the addition of the bias in the training phase (i.e. an increase in the number of parameters). This indicates that it is not overfitting the data, rather it is correctly reproducing the initial

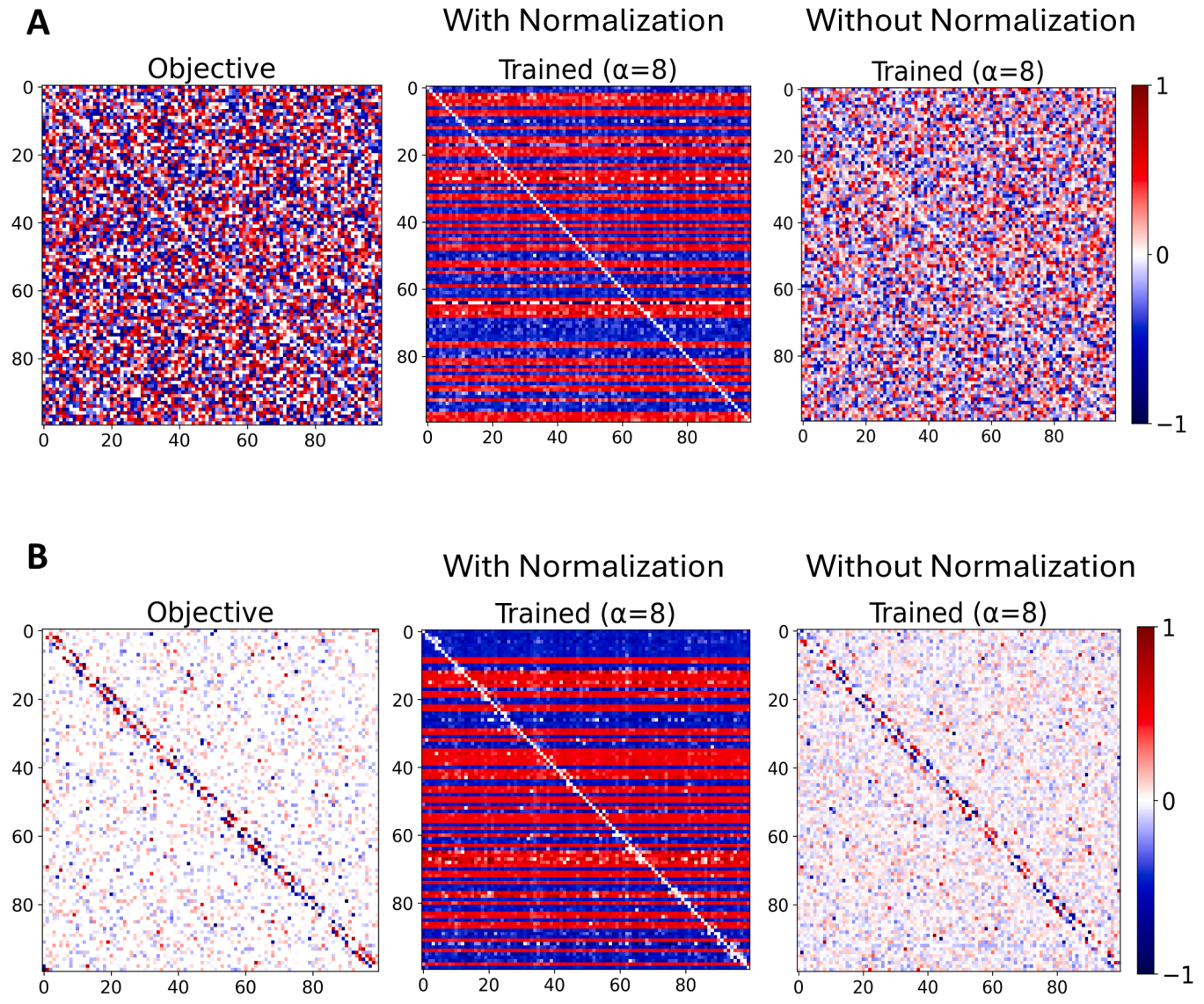

Fig. 9. Comparison between trained matrices obtained from the respective objective matrices with and without normalization, for a large learning rate (α=8). (A) Effects of normalization for the dense matrix. (B) Effects of normalization for the small world architecture.

objective matrices without the bias neuron.

To verify if the bias term is negligible, we analysed the distributions of the Root Mean Square Values (RMSVs) of the biases and we compared them with the threshold of the statistical significance of the estimated edges, computed as described in Fig. S25. We reported the results in Fig. S24 of the supplementary materials. We can observe that, for both the small world and the dense architecture, and for all the tested network sizes (N = 20-200), biases are on average smaller than the smallest significant weight and, consistently with the zero-bias generative process, they can be considered noise-level.

In conclusion, since the bias model does worsen, but does not meaningfully improve generalization either, we believe it is reasonable to choose the simpler model without bias optimization.

4.3. Experimental data

In this work, we applied RHoMM equation (Eq. 1) to synthetic effective connectivity matrices with different architectures to generate synthetic binary sequences and then we examined whether applying the RHoMM framework to the obtained binary sequences we could faithfully recover the ground-truth synthetic matrices. The objective was to verify the ability of the model to retrieve a target effective connectivity matrix and to analyse how different hyperparameters could affect this ability, with a ground-truth to compare the results with and a better control and understanding of data characteristics and variability.

However, relying exclusively on synthetic data to study the model's behaviour may present limitations. The use of simulated data, on one hand simplifies development but it risks overestimating model performance on experimental data. As a matter of facts, real MEG data can introduce damaging effects – such as noise, artifacts, volume conduction and nonstationarity – that can lead to distorted results or instability. On the other hand, an experimental dataset could also have beneficial effects (physiological constraints, variability, complexity) that can improve robustness and ecological validity of connectivity estimation.

To address this, we incorporated an analysis of an experimental MEG dataset, through which we extended the conclusions of our simulations to real data and of Gosti et al., 2024 to denser parcellation schemes.

4.3.1. Data binarization

The fundamental unit of RHoMM is the McCulloch-Pitts neuron, a binary computational unit that can assume only two discrete values: 1 for an active state and 0 for the rest condition. Each seed voxel is

associated with one these computational units, that act as the network's neurons. For this reason, in order to train the model using the experimental data, we had to transform the BLP continuous signals into binary time series. While the process of binarization does imply a loss of information, involving loss of amplitude details and a possible oversimplification of the temporal dynamics, it also results in a non-negligible improvement of the computational efficiency. For this reason, is often considered useful to model the data as point process (Cox and Isham, 1980), i.e. as a series of discrete events. Previous studies have already shown that the statistical properties of the dynamical regime can be effectively encapsulated within a point process representation (Castro et al., 1997). On this ground, Tagliazucchi et al. (2012) were able to demonstrate with fMRI that the BLP time course binarization most likely preserves the functional connectivity dynamics, and that the brain dynamics representation as a discrete process might facilitate large-scale analysis of brain functionality.

To verify that the binarization process does not affect the reciprocal statistical dependencies of the signal, we computed the mean functional connectivity of the continuous BLP and the binarized time series in the alpha and beta band, both for the 155-nodes parcellation and the 45-nodes parcellation. Specifically, we calculated for each run the Pearson's correlation matrix, for the continuous and binarized signals, and we normalized the correlation values applying a z-Fisher transformation. Through a Mantel test, we then compared the connectomes obtained from the continuous and the binarized BLP (Figs. S26 and S27). The results showed a highly significant correlation between the connectomes, with a Pearson's correlation coefficient close to one for both parcellation and bands, and a p-value < 0.001 ($r=0.99$ for 155-node parcellation and $r=0.98$ for the 45-nodes parcellation in both frequency bands).

This outcome strengthens the hypothesis that data binarization, even if it implies some loss of information and data simplification, allows the preservation of the BLP non linearity and the associated functional connectivity pattern.

4.4. Methodological considerations

The RHoMM is designed to be applied to source level MEG BLP timecourses, that have a temporal resolution ($\approx$ 10 ms) and a richness of information considerably higher than fMRI data ($\approx$ 1 s temporal resolution). This increase in the temporal resolution implies an increase in computational time, since the model is applied to a number of time points about 100 times larger.

The computational burden of the model has been previously discussed in Gosti et al. (2024), where they performed a simulation to understand how the execution time grows with the number of nodes. From this simulation, they estimated a time of approximately 9 seconds for the model to converge to a solution for networks comparable with functional connectomes ($N=150$) and for 350 transitions. They compared these results to those obtained with the regressive DCM and the standard DCM, when applied on fMRI data (Frässle, 2017; Frässle, 2018; Frässle, 2021). From this comparison the execution rate of RHoMM was found competitive, if not better than both classic DCM and regressive DCM.

Following the same procedure to verify the scalability to large network sizes, we performed some simulations to understand how the execution time grows with the number of nodes. As described in Gosti et al. (2024), the abundance of temporal samples (about 14700 samples for ~5 minutes recordings) allows to divide the recordings into segments of n binary vectors (transitions) and estimate the mean effective connectivity from them. Following the simulation results we selected $n=700$ as the optimal number of transitions. To provide predictions on the computational burden for different network sizes, we therefore used simulated runs of 700 binary vectors. We simulated random graph networks with edge weights randomly drawn from a uniform distribution $[-1,1]$ and set to zero the weights below a threshold of 0.02. Based on the simulations results, we selected the suitable hyperparameters ensuring convergence for the different node sizes ($N\text{steps}=4000$ and $\alpha=4$). We ran the simulations on an 8-cores CPU using 30 runs of 700 transitions, and we estimated the mean execution time for each network size.

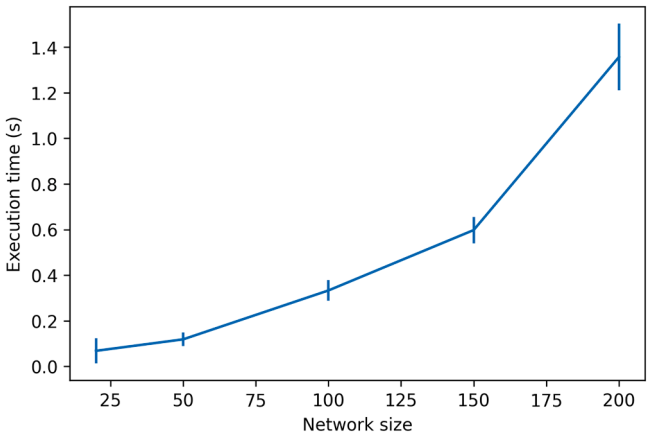

Fig. 10. Computational burden. This plot shows an estimate of the computational burden when scaling RHoMM. It shows the execution time for simulated dynamics obtained from synthetic matrices with different sizes (N=20, 50, 100, 150, 200).

In line with the results obtained in Gosti et al. (2024), the trend of the obtained curve is not linear (see Fig. 10) and the simulations we performed revealed an even more encouraging outcome. As we can observe in Fig. 10, we found that for a 200-nodes dense network (without normalization and without bias), with 700 transitions and an optimal learning rate $\alpha=4$, the model reached convergence in about 1.4 seconds. In this perspective, if we consider a BLP time series composed of about 20 segments of 700 transitions (approximately 5 minutes recording), we can estimate that the model would take about 30 seconds to converge to a solution.

On this ground, although, as already pointed out in Gosti et al. (2024), the number of iterations has a non-negligible correlation with the convergence to the optimal solution, we can still reinforce the statement of RHoMM execution rate competitiveness.

Another interesting point to discuss is the significance of the weights. Since our main purpose in this work is to evaluate the RHoMM on a network level, and not on a nodal level, for this whole performance analysis, we considered a global metric (i.e. the MSE between the objective and estimated matrices), that do not provide information on the edges individually. When we move onto real MEG data, though, we may be interested in understanding the significance of the connections between different brain areas. With this purpose, we demonstrated that it is possible to establish for the estimated matrix a threshold of the statistical significance for the single edges. We employed the same method carried out in Gosti et al, 2024 (Appendix C), in order to validate their results also on larger networks, obtained without the normalization in the training. For a more in-depth explanation refer to the supplementary materials (Fig. S25).

5. Conclusions

In this work we evaluated the scalability of the Recurrent Hopfield Mass Model to large networks (150-200 nodes) using simulated data, together with the effect of optimization strategies, such as normalization of the weights at each GD step, and bias addition in the training process. Specifically, since RHoMM has already been tested on connectomes with less than 50 nodes (Gosti, 2024), we evaluated the model's outcomes when applied on larger networks while tuning the hyperparameters to optimize its performance, to understand how critical these are in the

perspective of applying RHoMM to real MEG data. The results were independent of the simulated architecture and showed significant improvements both on the model convergence, widening the learning rate convergence interval, and performance following the exclusion of the normalization from the training, while the addition of the bias had no significant impact on the model outcome. These results lead to the conclusion that when applying RHoMM to real data, adopting the version without weight normalization would make the selection of the learning rate less critical than the version used in (Gosti, 2024), promoting the use of larger learning rates with a consequent decrease of the computational cost.

After selecting the optimal training configuration and fine-tuning the hyperparameters, we applied the model on experimental MEG data with two different parcellation schemes (45 and 155 nodes). The results obtained led us to test the hypothesis that the model can reliably estimate the effective connectivity matrix underlying cerebral binarized power time courses and generate binary time series with the same functional properties of the experimental ones. This evidence, already pointed out in Gosti et al, 2024 for networks of 45 nodes, can now be extended to real effective MEG connectomes with denser parcellation (comparable with functional connectomes).

In summary, RHoMM, being simple and scalable to encode both small and large-scale systems interactions on a millisecond-resolution time scale (MEG BLP time series), could be suitable for extensive applications in the field of system neuroscience.

CRediT authorship contribution statement

Martina Ferrazza: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Giorgio Gosti:** Writing – review & editing, Visualization, Supervision, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Edoardo Milanetti:** Writing – review & editing, Software, Methodology, Conceptualization. **Sara Spadone:** Writing – review & editing, Resources, Conceptualization. **Giancarlo Ruocco:** Writing – review & editing, Methodology, Conceptualization. **Stefania Della Penna:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Martina Ferrazza is supported by the National PhD Program TAN with contribution from the University of Chieti- Pescara, PNRR Dr. 351, 38° cycle.

Giorgio Gosti wishes to acknowledge financial support H2IOSC Project- Humanities and cultural Heritage Italian Open Science Cloud funded by the European Union – NextGenerationEU – NRRP M4C2 - Project code IR0000029- CUP B63C22000730005.

Stefania Della Penna was supported by the European Union - Next Generation EU, Mission 4, Component 2, CUP C53D23008470001.

Edoardo Milanetti was supported by MUR PRIN 2022 (CUP B53D2300399 0006).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.neunet.2025.108406.

References

Allen, E. A., Damaraju, E., Plis, S. M., Erhardt, E. B., Eichele, T., & Calhoun, V. D. (2014). Tracking whole-brain connectivity dynamics in the resting state. *Cerebral Cortex, 24* (3), 663–676. https://doi.org/10.1093/cercor/bhs352, 2014 MarEpub 2012 Nov 11. PMID: 23146964; PMCID: PMC3920766.

Amit, D. J., Gutfreund, H., & Sompolinsky, H. (1985). Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters, 55*(14), 1530.

Baker A.P., Brookes M.J., Rezek I.A., Smith S.M., Behrens T., Smith P.J.P., Woolrich M. (2014). Fast transient networks in spontaneous human brain activity eLife 3:e01867. https://doi.org/10.7554/eLife.01867.

Bassett, D. S., & Sporns, O. (2017). Network neuroscience. *Nature Neuroscience, 20*(3), 353–364.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research, 13*(10), 281–305.

Betzel, R. F., & Bassett, D. S. (2017). Generative models for network neuroscience: prospects and promise. *Journal of the Royal Society Interface, 14*. Article 20170623.

Betzel, R. F., Avena-Koenigsberger, A., Goñi, J., He, Y., de Reus, M. A., Griffa, A., et al. (2016). Generative models of the human connectome. *NeuroImage, 124*, 1054–1064.

Biswas, A. (2024). Hidden Synergy: L_1 Weight Normalization and 1-Path-Norm Regularization No. arXiv:2404.19112. 10.48550/arXiv.2404.19112.

Bottou, L. (2012). Stochastic gradient descent tricks. *Neural Networks, Tricks of the Trade, Reloaded,, 7700*, 430–445.

Brandes, U., Gaertler, M., & Wagner, D. (2003). Experiments on graph clustering algorithms. *LNCS*, 2832. https://doi.org/10.1007/978-3-540-39658-1_52

Breakspear, M. (2017). Dynamic models of large-scale brain activity. *Nature Neuroscience, 20*, 340–352.

Bullmore, E., & Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews Neuroscience, 10*(3), 186–198. https://doi.org/10.1038/nrn2575

Cabral, J., Hugues, E., Sporns, O., & Deco, G. (2011). Role of local network oscillations in resting-state functional connectivity. *NeuroImage, 57*, 130–139.

Cabral, J., Kringelbach, M. L., & Deco, G. (2014a). Exploring the network dynamics underlying brain activity during rest. *Progress in Neurobiology, 114*, 102–131.

Cabral, J., Luckhoo, H., Woolrich, M., Joensson, M., Mohseni, H., Baker, A., et al. (2014b). Exploring mechanisms of spontaneous functional connectivity in MEG: how delayed network interactions lead to structured amplitude envelopes of band-pass filtered oscillations. *NeuroImage, 90*, 423–435.

Carpenter, G. A. (1989). Neural network models for pattern recognition and associative memory. *Neural Networks, 2*(4), 243–257.

Castro, R., & Sauer, T. (1997). Correlation dimension of attractors through interspike intervals. *Physical Review E, 55*(1), 287–290. https://doi.org/10.1103/physreve.55.287

Chen, R., Singh, M., Braver, T. S., & Ching, S. (2025). Dynamical models reveal anatomically reliable attractor landscapes embedded in resting-state brain networks. *Imaging Neuroscience*. https://doi.org/10.1162/imag_a_00442, 2025;3:imag_a_00442Epub 2025 Jan 24. PMID: 40475237; PMCID: PMC12140615.

Cox, D. R., & Isham, V. (1980). *Point Processes* (1st ed.). Routledge. https://doi.org/10.1201/9780203743034

David, O., Kiebel, S. J., Harrison, L. M., Mattout, J., Kilner, J. M., & Friston, K. J. (2006). Dynamic causal modeling of evoked responses in EEG and MEG. *NeuroImage, 30*(4), 1255–1272.

de Pasquale, F., Della Penna, S., Snyder, A. Z., Marzetti, L., Pizzella, V., Romani, G. L., et al. (2012). A cortical core for dynamic integration of functional networks in the resting human brain. *Neuron, 74*(4), 753–764.

de Pasquale, F., Della Penna, S., Sporns, O., Romani, G. L., & Corbetta, M. (2016). A dynamic core network and global efficiency in the resting human brain. *Cerebral Cortex, 26*(10), 4015–4033.

de Pasquale, F., Corbetta, M., Betti, V., & Della Penna, S. (2018). Cortical cores in network dynamics. *NeuroImage,, 180*, 370–382.

de Pasquale, F., Spadone, S., Betti, V., Corbetta, M., & Della Penna, S. (2021). Temporal modes of hub synchronization at rest. *NeuroImage, 235*. Article 118005.

Deco, G., Jirsa, V. K., & McIntosh, AR. (2013). Resting brains never rest: computational insights into potential cognitive architectures. *Trends in Neurosciences, 36*(5), 268–274.

Deco, G., Cruzat, J., Cabral, J., Tagliazucchi, E., Laufs, H., Logothetis, N. K., & Kringelbach, M. L. (2019). Awakening: predicting external stimulation to force transitions between different brain states. *Proceeding for National Academy of Science USA, 116*(36), 18088–18097. https://doi.org/10.1073/pnas.1905534116

Della Penna, S., Delgratta, C., Granata, C., Pasquarelli, A., Pizzella, V., Rossi, R., et al. (2000). Biomagnetic systems for clinical use. *Philosophical Magazine B, 80*(5), 937–948.

Englert, R., Kincses, B., Kotikalapudi, R., Gallitto, G., Li, J., Hoffschlag, K., Choong-Wan, Woo, Wager Tor, D., Timmann, D., Bingel, U., & Spisak, T. (2024). Connectome-based attractor dynamics underlie brain activity in rest, task, and disease. *eLife, 13*. https://doi.org/10.7554/eLife.98725.1. RP98725.

Fanini, B., & Gosti, G. (2024). A new generation of collaborative immersive analytics on the web: open-source services to capture, process and inspect users' Sessions in 3D environments. *Future internet, 16*(5), 147.

Fanini, B., Ferdani, D., Demetrescu, E., Berto, S., & d'Annibale, E. (2021). ATON: an open-source framework for creating immersive, collaborative and liquid web-apps for cultural heritage. *Applied Sciences, 11*(22), Article 11062.

Folli, V., Gosti, G., Leonetti, M., & Ruocco, G. (2018). Effect of dilution in asymmetric recurrent neural networks. *Neural Networks, 104*, 50–59.

Fortunato, S. (2009). Community detection in graphs. *Physical Reports, 486*(3–5), 75–174. VolumeIssue https://arxiv.org/abs/0906.0612.
Frässle, S., Lomakina, E. I., Razi, A., Friston, K. J., Buhmann, J. M., & Stephan, K. E. (2017). Regression DCM for fMRI. *NeuroImage,, 155*, 406–421.
Frässle, S., Lomakina, E. I., Kasper, L., Manjaly, Z. M., Leff, A., Pruessmann, K. P., et al. (2018). A generative model of whole-brain effective connectivity. *NeuroImage, 179*, 505–529.
Frässle, S., Harrison, S. J., Heinzle, J., Clementz, B. A., Tamminga, C. A., Sweeney, J. A., et al. (2021). Regression dynamic causal modeling for resting-state fMRI. *Human Brain Mapping, 42*(7), 2159–2180.
Friston, K. J., Li, B., Daunizeau, J., & Stephan, K. E. (2011b). Network discovery with DCM. *NeuroImage,, 56*(3), 1202–1221.
Friston, K. J., Kahan, J., Biswal, B., & Razi, A. (2014). A DCM for resting state fMRI. *NeuroImage, 94*, 396–407.
Friston, K. J. (2011a). Functional and effective connectivity: A review. *Brain Connectivity, 1*(1), 13–36.
Gosti, G., Folli, V., Leonetti, M., & Ruocco, G. (2019). Beyond the maximum storage capacity limit in hopfield recurrent neural networks. *Entropy, 21*(8), 726.
Gosti, G., Milanetti, E., Folli, V., de Pasquale, F., Leonetti, M., Corbetta, M., & Della Penna, S. (2024). A recurrent Hopfield network for estimating meso-scale effective connectivity in MEG. *Neural Networks, 170*, 72–93.
Grossberg, S., & Levine, D. (1975). Some developmental and attentional biases in the contrast enhancement and short term memory of recurrent neural networks. *Journal of Theoretical Biology, 53*, 341–380.
Grossberg, S. (2007). Towards a unified theory of neocortex: laminar cortical circuits for vision and cognition. *Progress in Brain Research, 165*, 79–104.
Hahn, G., Skeide, M. A., Mantini, D., Ganzetti, M., Destexhe, A., Friederici, A. D., et al. (2019). A new computational approach to estimate whole-brain effective connectivity from functional and structural MRI, applied to language development. *Scientific Reports, 9*(1), 8479.
Haykin, S. (2009). *Neural networks and learning machines*.
Hodgkin, A. L., & Huxley, A. F. (1952). Propagation of electrical signals along giant nerve fibres. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 140*, 177–183.
Hoffer, E., Banner, R., Golan, I., & Soudry, D. (2019). Norm matters: efficient and accurate normalization schemes in deep networks. *Advances in Neural Information Processing Systems, 31*.
Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, 79*(8), 2554–2558.
Hutchison, R. M., Womelsdorf, T., Allen, E. A., Bandettini, P. A., Calhoun, V. D., Corbetta, M., Della Penna, S., Duyn, J. H., Glover, G. H., Gonzalez-Castillo, J., Handwerker, D. A., Keilholz, S., Kiviniemi, V., Leopold, D. A., de Pasquale, F., Sporns, O., Walter, M., & Chang, C. (2013). Dynamic functional connectivity: promise, issues, and interpretations. *NeuroImage, 80*, 360–378. https://doi.org/10.1016/j.neuroimage.2013.05.079, 2013 Oct 15Epub 2013 May 24. PMID: 23707587; PMCID: PMC3807588.
Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456).
Jafarian, A., Litvak, V., Cagnan, H., Friston, K. J., & Zeidman, P. (2020). Comparing dynamic causal models of neurovascular coupling with fMRI and EEG/MEG. *NeuroImage, 216*. Article 116734.
Kiebel, S. J., Garrido, M. I., Moran, R. J., & Friston, K. J. (2008). Dynamic causal modelling for EEG and MEG. *Cognitive Neurodynamics, 2*, 121–136.
Kruskal, W. H., & Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association, 47*(260), 583–621. https://doi.org/10.1080/01621459.1952.10483441
Laarhoven, T. van. (2017). L2 Regularization versus Batch and Weight Normalization (No. arXiv:1706.05350). 10.48550/arXiv.1706.05350.
Lansner, A. (2009). Associative memory models: from the cell-assembly theory to biophysically detailed cortex simulations. *Trends in Neurosciences, 32*(3), 178–186.
Leonetti, M., Folli, V., Milanetti, E., Ruocco, G., & Gosti, G. (2020). Network dilution and asymmetry in an efficient brain. *Philosophical Magazine, 100*(20), 2544–2555.
Li, X., Chen, S., & Yang, J. (2020). Understanding the Disharmony between Weight Normalization Family and Weight Decay. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(04), 4715–4722. https://doi.org/10.1609/aaai.v34i04.5904
Liao, R., Xiong, Y., Fetaya, E., Zhang, L., Yoon, K., Pitkow, X., & Zemel, R. (2019). Reviving and improving recurrent back-propagation. In *International conference on machine learning* (pp. 3082–3091). PMLR.
Liu, Z., Cui, Y., Wan, J., Mao, Y., & Chan, A.B. (2022). Weight Rescaling: Effective and Robust Regularization for Deep Neural Networks with Batch Normalization (No. arXiv:2102.03497). 10.48550/arXiv.2102.03497.
Lu, Q., Li, H., Luo, G., Wang, Y., Tang, H., Han, L., et al. (2012). Impaired prefrontal–amygdala effective connectivity is responsible for the dysfunction of emotion process in major depressive disorder: A dynamic causal modeling study on MEG. *Neuroscience Letters, 523*(2), 125–130.
Connectivity inference from neural recording data: challenges, mathematical bases and research directionsMagrans de Abril, I., Yoshimoto, J., & Doya, K. (Eds.), Connectivity inference from neural recording data: challenges, mathematical bases and research directions. *Neural Networks, 102*, (2018), 120–137.
Mantel, N. (1967). The detection of disease clustering and a generalized regression approach. *Cancer Research, 27*(2), 209–220.
Mantini, D., Penna, S. D., Marzetti, L., de Pasquale, F., Pizzella, V., Corbetta, M., et al. (2011). A signal-processing pipeline for magnetoencephalography resting-state networks. *Brain Connectivity, 1*(1), 49–59.
Montazeri, A., & Schmidt, R. (2024). Robustness in Hopfield neural networks with biased memory patterns. *bioRxiv : the Preprint Server for Biology*, 2024-10.
Rubinov, M., Knock, S. A., Stam, C. J., Micheloyannis, S., Harris, A. W., Williams, L. M., & Breakspear, M. (2009). Small-world properties of nonlinear brain activity in schizophrenia. *Human Brain Mapping, 30*(2), 403–416. https://doi.org/10.1002/hbm.20517
Salimans, T., & Kingma, D.P. (2016). Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. 10.48550/arXiv.1602.07868 (No. arXiv:1602.07868).
Singh, M. F., Braver, T. S., Cole, M. W., & Ching, S. N. (2020). Estimation and validation of individualized dynamic brain models with resting state fMRI. *NeuroImage, 221*. Article 117046.
Tagliazucchi, E., Balenzuela, P., Fraiman, D., & Chialvo, D. R. (2012). Criticality in Large-Scale Brain fMRI Dynamics Unveiled by a Novel Point Process Analysis. *Frontiers in Physiology, 3*. https://doi.org/10.3389/fphys.2012.00015
Vidaurre, D., Smith, S. M., & Woolrich, M. W. (2017). Brain network dynamics are hierarchically organized in time. *Proceeding for National Academy of Science USA, 114*(48), 12827–12832. https://doi.org/10.1073/pnas.1705120114
Watanabe, T., Masuda, N., Megumi, F., Kanai, R., & Rees, G. (2014). Energy landscape and dynamics of brain activity during human bistable perception. *Nature Communications, 5*, 4765. https://doi.org/10.1038/ncomms5765, 2014 Aug 28PMID: 25163855; PMCID: PMC4174295.
Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE, 78*(10), 1550–1560.
Yuan, Q., & Xiao, N. (2019). Scaling-Based Weight Normalization for Deep Neural Networks. *IEEE Access, 7*, 7286–7295. https://doi.org/10.1109/ACCESS.2018.2890373
Zhang, S., Jiang, H., Wei, S., & Dai, L.-R. (2015). Rectified linear neural networks with tied-scalar regularization for LVCSR. In *Proceedings of the annual conference of the international speech communication association* (pp. 2635–2639). vol. 2015-JanuaInternational Speech and Communication Association.