Functional Connectomes of Neural Networks

Tananun Songdechakraiwut, Yutong Wu

Department of Computer Science, Duke University
t.song@duke.edu, mason.wu@duke.edu

Abstract

The human brain is a complex system, and understanding its mechanisms has been a long-standing challenge in neuroscience. The study of the functional connectome, which maps the functional connections between different brain regions, has provided valuable insights through various advanced analysis techniques developed over the years. Similarly, neural networks, inspired by the brain's architecture, have achieved notable success in diverse applications but are often noted for their lack of interpretability. In this paper, we propose a novel approach that bridges neural networks and human brain functions by leveraging brain-inspired techniques. Our approach, grounded in the insights from the functional connectome, offers scalable ways to characterize topology of large neural networks using stable statistical and machine learning techniques. Our empirical analysis demonstrates its capability to enhance the interpretability of neural networks, providing a deeper understanding of their underlying mechanisms.

Code — https://github.com/masonwu11/topo-fcnn
Extended version — https://arxiv.org/abs/2412.15279

1 Introduction

The human brain is an incredibly complex system, and understanding its intricate workings has been a long-standing challenge in neuroscience. One well-established approach to gaining deeper insights into the brain's underlying mechanisms is through the study of the functional connectome, which maps the functional connections between regions of brain networks and reflects the brain's dynamic network of interactions (Bullmore and Sporns 2009). In recent decades, successful findings have emerged from this field, thanks to the development of a wide array of analysis techniques.

Artificial neural networks, inspired by the architecture and functioning of the human brain, have achieved remarkable success in applications ranging from image recognition (He et al. 2016) to natural language processing (Vaswani et al. 2017). However, despite these successes, neural networks are often considered black box models due to their lack of interpretability and the difficulty in understanding the underlying mechanisms driving their performance. Neural networks have predesigned architectures with preconfigured weights connecting neurons, analogous to how physically connected brain regions with white matter fibers provide structural information measured by diffusion MRI. Similarly, functional connections between distant neurons resemble how functional MRI measures functional connectivity between brain regions that may not have direct neuroanatomical connections. These connections in the brain give rise to coordinated activity patterns crucial for cognitive processes (Honey et al. 2009). Given that neural networks are simplified artificial versions of brain functions, it stands to reason that insights from the functional connectome could be leveraged to enhance our understanding, interpretability, and analysis of these networks, potentially leading to the development of more transparent and efficient models.

However, analyzing functional connectomes is inherently challenging due to the need to extract subtle topological patterns from noisy, complete graphs. Therefore, typical workflows apply a threshold to obtain a sparser graph with a clearer structure before applying techniques from graph theory (Bullmore and Sporns 2009). Graph theory has played a crucial role in functional connectome research; however, prior analyses utilizing graph theory have primarily focused on pairwise dyadic relationships, often at a fixed spatial threshold. This approach, centered on dyads, limits the neural structures and functions that graph theory can investigate. Given that a neural network processes information not only based on local neurons of a subnetwork but also across the entire network, from input to output layers, a more comprehensive understanding requires a shift in perspective–from pairwise interactions to capturing higher-order relations (topology) across the full range of spatial resolutions.

Persistent homology (Edelsbrunner and Harer 2022), an algebraic topology technique, has emerged as a promising tool for understanding and quantifying the topology of the human brain (Sizemore et al. 2018; Songdechakraiwut, Shen, and Chung 2021). Recently, there have been increasing attempts to apply persistent homology to study the interpretability of deep learning. These studies suggest that persistent homology can extract high-order topological information to interpret neural networks, but challenges remain. In particular, current methods often focus solely on the network's structural information, without incorporating data-driven tasks (Rieck et al. 2019; Watanabe and Yamana

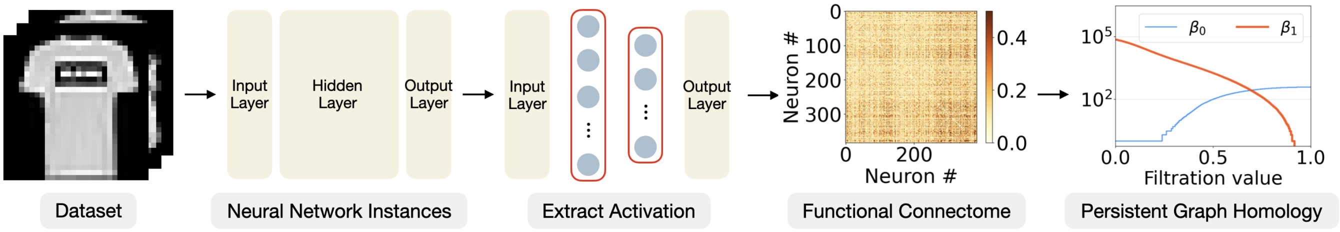

Figure 1: A schematic for extracting persistent graph homology, representing the topology of neural-network-derived functional connectomes.

2022), and are typically limited to models with a small number of neurons (Naitzat, Zhitnikov, and Lim 2020; Zhang et al. 2023). Notably, Zhang et al. has applied persistent homology to study the functional connectivities of neural networks. However, their approach is still limited by the cubic time complexity of persistent homology techniques (Otter et al. 2017), making them computationally infeasible for large, complete graphs, and relies heavily on approximation solutions. The authors addressed this challenge by thresholding functional connectivities at a predefined threshold, which limits the spatial resolution of their analysis to a range of pre-determined values, rather than capturing the entire global mechanisms of neural network functions. Importantly, varying threshold values can significantly alter the graph structure, affecting its robustness and sensitivity to signals, and potentially influencing the study's final outcome.

Persistent graph homology (Songdechakraiwut and Chung 2023), a recent persistent homology advancement inspired by the human brain, captures interpretable topological invariants of great interest in functional brain networks, including connected components and cycles, across the *entire* spatial scale of a graph. Connected components characterize the network shape (Tewarie et al. 2015), while cycles represent higher-order interactions potentially associated with information propagation and feedback control (Kwon and Cho 2007). The computability of persistent graph homology enables the analysis of intricate details in large-scale, complex brain networks. Motivated by this, we adopt a brain-inspired graph homology perspective to overcome computational limitations and fully exploit the potential of topological analyses on large-scale neural network functions.

In this paper, we propose a novel analysis framework that bridges neural networks and human brain functions by leveraging brain-inspired techniques from functional MRI and persistent graph homology. Our framework, grounded in insights from the functional connectome, addresses the aforementioned challenges and offers *scalable* methods to characterize the topology of functional mechanisms of *large* neural networks using *stable* statistical and machine learning techniques. We define functional connectomes, a representation used to describe functions in neural networks, without the need for predefined threshold values, thereby significantly improving the scientific rigor of the analysis. We also demonstrate the computability and efficiency of statistics for such representations via analytic forms. Specifically, we present closed-form computations for the Wasserstein distance, Wasserstein statistics–including barycenter (average) and variance–and the Wasserstein gradient, enabling the realization of the robustness benefits of true Wasserstein distance, grounded in the central stability theorem (Skraba and Turner 2023). Utilizing these computable statistics naturally leads to the development of a centroid-based clustering strategy, where the Wasserstein barycenter serves as the topological centroid. We conducted extensive experiments using this method to validate that our framework can indeed enhance our ability to discern and interpret the complex structure of neural network functions, providing new avenues for both analysis and application.

2 Functional Connectome Framework

Figure 1 shows a schematic of the functional connectivity analysis framework, described in the following.

Functions of Neural Networks

Given a collection of data samples, we partition it into two separate datasets: a training dataset and a functional dataset. The training dataset is utilized via k-fold cross-validation to determine a set of optimal hyperparameter values through grid search. Using these optimal values, we train a well-generalized neural network on the entire training data. Subsequently, the functional dataset is fed into the fully-trained neural network, resulting from the aforementioned training process, to investigate the information propagation of new, unknown data through it.

Without loss of generality, we will assume that the neural network architecture of interest is a feedforward network. Given a fully-trained feedforward network with M hidden neurons and a functional dataset denoted by $\mathbb{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$, each data point $\boldsymbol{x}^{(i)} \in \mathbb{X}$ is inputted into the feedforward network and processed through a series of neurons. Each j-th neuron uses an affine transformation followed by an activation function to produce its output a_{ij}, representing the information processing for $\boldsymbol{x}^{(i)}$. By processing the entire functional dataset, we concatenate the j-th neuron outputs for all the data points into a *functional vector* $\boldsymbol{a}_j = (a_{1j}, a_{2j}, \ldots, a_{Nj})$, representing the functional signal in a neural network, similar to the functions of connectome circuits studied in functional MRI.

When two functional vectors from a pair of neurons are statistically dependent, they exhibit functional synergy. To quantify this synergy, several methods can be used to measure the statistical dependency between the two vectors, such as Pearson correlation, partial correlation, Spearman's rank correlation, and mutual information (Fornito, Zalesky, and Bullmore 2016). In macroscale connectomics, Pearson correlation, which captures linear relationships between different brain regions, is widely used for computing functional connectomes of the human brain. It is straightforward to interpret, with values ranging from -1 to 1, where values further from 0 indicate stronger functional connections. Pearson correlation effectively detects synchrony between brain regions, a common feature of neural activity. For similar reasons, we will use Pearson correlation in this paper to analyze neural networks. Additionally, Pearson correlation's computational efficiency makes it feasible to apply to large neural network models that involve deep layers and many neurons. Formally, Pearson correlation between two functional vectors $\boldsymbol{a}_j = (a_{1j}, a_{2j}, ..., a_{Nj})$ and $\boldsymbol{a}_k = (a_{1k}, a_{2k}, ..., a_{Nk})$ is defined as

$$\rho_{jk} = \frac{\sum_{i=1}^{N}(a_{ij} - \overline{\boldsymbol{a}}_j)(a_{ik} - \overline{\boldsymbol{a}}_k)}{\sqrt{\sum_{i=1}^{N}(a_{ij} - \overline{\boldsymbol{a}}_j)^2}\sqrt{\sum_{i=1}^{N}(a_{ik} - \overline{\boldsymbol{a}}_k)^2}},$$

where $\overline{\boldsymbol{a}}_j = \sum_{i=1}^{N} a_{ij}/N$ and $\overline{\boldsymbol{a}}_k = \sum_{i=1}^{N} a_{ik}/N$.

In studies of the connectome of the human brain, a connectome is typically represented as a graph, comprising brain regions of interest as nodes, and pairwise correlations as edge weights (Fornito, Zalesky, and Bullmore 2016). There is no universally accepted standard for whether the sign of the correlations should be preserved. Generally, choosing the absolute value of negative correlations highlights the strength of connectivity without regard to its direction. This can be particularly useful in analyses where the primary interest is in the magnitude of interactions between brain regions, regardless of whether they are positively or negatively correlated. Additionally, Pearson correlation between the same region is always 1, and thus is typically excluded from analyses, resulting in a brain graph with no self-loops. For similar reasons, we will follow the same well-established procedure applied to the case of a connectome of a neural network. Formally, given the correlations between every pair of hidden neurons, we define an M-by-M weighted adjacency matrix $\boldsymbol{G}$ whose jk-th entry is given as

$$G_{jk} = \begin{cases} |\rho_{jk}| & \text{if } j \neq k; \\ 0 & \text{otherwise.} \end{cases}$$

We will call this matrix a *functional connectome* of a neural network.

Persistent Graph Homology

The functional connectome is a very dense matrix, typically representing a *complete* graph. Therefore, typical workflows (Bullmore and Sporns 2009) apply a threshold to the correlation values for two main reasons: to obtain a sparser graph with a more apparent structure and to reduce the time complexity of computational methods. In particular, methods in persistent homology have cubic time complexity (Otter et al. 2017), making them computationally infeasible for large, complete graphs. This requires iterative, approximation solutions, which introduce numerical errors and reduce the signal-to-noise ratio. Additionally, varying different threshold values significantly alters the graph structure, potentially affecting the study's final outcome. Importantly, pairwise dyadic correlations at a fixed spatial threshold constrain the neural structures and functions that can be investigated. Given that a neural network processes information not only based on local neurons of a subnetwork but also the entire neural network from input to output layers, a more comprehensive understanding requires a shift in perspective from pairwise interactions to capturing higher-order relations across the entire range of spatial resolutions.

In this work, we address these challenges by leveraging brain-inspired *persistent graph homology*, a *scalable* topological-learning paradigm that enables analyses of large-scale functional connectomes *without* approximation (Songdechakraiwut et al. 2023). It has emerged as a promising tool for understanding, characterizing, and quantifying human connectomes (Songdechakraiwut et al. 2022). Persistent graph homology describes interpretable topological invariants, including connected components (0th homology group) and independent cycles (1st homology group or cycle rank), across the *entire* spatial scale of a graph.

Formally, given a functional connectome $\boldsymbol{G}$, we define a binary graph $\boldsymbol{G}_\epsilon$ with the same set of neurons by thresholding the edge correlations so that an edge between neurons j and k exists if $\rho_{jk} > \epsilon$. As ϵ increases, more edges are removed from the functional connectome $\boldsymbol{G}$. Thus, we have a filtration (Lee et al. 2012): $\boldsymbol{G}_{\epsilon_0} \supseteq \boldsymbol{G}_{\epsilon_1} \supseteq \cdots \supseteq \boldsymbol{G}_{\epsilon_k}$, where $\epsilon_0 \leq \epsilon_1 \leq \cdots \leq \epsilon_k$ are called filtration values. Persistent homology tracks the birth and death of the topological invariants over these filtration values ϵ. A topological invariant born at filtration b_j and persisting up to filtration d_j is represented by a point (b_j, d_j) in a 2D plane. The set of all such points (b_j, d_j) is called a *persistence diagram* (Edelsbrunner and Harer 2022). As ϵ increases, the number of connected components $\beta_0(\boldsymbol{G}_\epsilon)$ increases monotonically, while the number of cycles $\beta_1(\boldsymbol{G}_\epsilon)$ decreases monotonically. Thus, persistent graph homology only needs to track a collection of sorted birth values $B(\boldsymbol{G})$ for the connected components and a collection of sorted death values $D(\boldsymbol{G})$ for the cycles, given as (Songdechakraiwut and Chung 2023)

$$B(\boldsymbol{G}) = \{b_j\}_{j=1}^{M-1}, \quad D(\boldsymbol{G}) = \{d_j\}_{j=1}^{1+M(M-3)/2}.$$

Figure 1 illustrates a schematic for extracting persistent graph homology, which represents the topology of functional connectomes derived from neural networks.

Scalability The set $B(\boldsymbol{G})$ consists of edge correlations found in the maximum spanning tree (MST) of $\boldsymbol{G}$. Once $B(\boldsymbol{G})$ is determined, the set $D(\boldsymbol{G})$ is derived from the edge correlations that are not part of the MST. Therefore, both $B(\boldsymbol{G})$ and $D(\boldsymbol{G})$ can be computed very efficiently in $O(n \log n)$ time, where n is the number of edges in the connectome graph.

Persistence Statistics

Distances are fundamental in statistics because they provide a way to measure how much individual data points vary, enabling the calculation of central tendency, dispersion, and overall data behavior. The Wasserstein distance is a prominent measure in persistent homology, associated with the central concept of the stability theorem (Skraba and Turner 2023). In this section, we will elaborate on the high computability of persistent-graph-homology-based Wasserstein distance and how it results in defining essential statistics such as the mean and variance, among others, which have potential in neural network interpretation.

Specifically, the Wasserstein distance between sets of birth values (or between sets of death values) can be obtained using a closed-form solution. Let $\boldsymbol{G}^{(1)}$ and $\boldsymbol{G}^{(2)}$ be two given functional connectomes, each having the same number of neurons. Then, the *exact* computation of the p-Wasserstein distance is achieved as (Songdechakraiwut and Chung 2023):

$$W_{p,B}(\boldsymbol{G}^{(1)}, \boldsymbol{G}^{(2)}) = \Big(\sum_{b \in B(\boldsymbol{G}^{(1)})} |b - \tau_0^*(b)|^p \Big)^{1/p},$$

$$W_{p,D}(\boldsymbol{G}^{(1)}, \boldsymbol{G}^{(2)}) = \Big(\sum_{d \in D(\boldsymbol{G}^{(1)})} |d - \tau_1^*(d)|^p \Big)^{1/p},$$

where τ_0^* maps the l-th smallest birth value in $B(\boldsymbol{G}^{(1)})$ to the l-th smallest birth value in $B(\boldsymbol{G}^{(2)})$, and τ_1^* maps the l-th smallest death value in $D(\boldsymbol{G}^{(1)})$ to the l-th smallest death value in $D(\boldsymbol{G}^{(2)})$, for all l. The exact Wasserstein distances $W_{p,B}$ and $W_{p,D}$ are well-defined because the bijective mappings τ_0^* and τ_1^* are well-defined for sets of births and deaths, respectively, with the same cardinality.

Importantly, the analytic expression of the Wasserstein distance above can be equivalently written in a more familiar Euclidean space. To do this, we define a vector of sorted birth values $\boldsymbol{b}_{\boldsymbol{G}^{(i)}} = (b_{i1}, b_{i2}, ..., b_{i,M-1})$ for the connected components, where $b_{ij} \in B(\boldsymbol{G}^{(i)})$ and $b_{ij} \leq b_{i,j+1}$. Similarly, we define a vector of sorted death values $\boldsymbol{d}_{\boldsymbol{G}^{(i)}} = (d_{i1}, d_{i2}, ..., d_{i,1+M(M-3)/2})$ for the cycles. With these definitions, the p-Wasserstein distance can be equivalently expressed as

$$W_{p,B}(\boldsymbol{G}^{(1)}, \boldsymbol{G}^{(2)}) = ||\boldsymbol{b}_{\boldsymbol{G}^{(1)}} - \boldsymbol{b}_{\boldsymbol{G}^{(2)}}||_p,$$

$$W_{p,D}(\boldsymbol{G}^{(1)}, \boldsymbol{G}^{(2)}) = ||\boldsymbol{d}_{\boldsymbol{G}^{(1)}} - \boldsymbol{d}_{\boldsymbol{G}^{(2)}}||_p,$$

where $||\cdot||_p$ is the L^p norm. Since $W_{p,B}$ and $W_{p,B}$ are differentiable functions and can be explicitly written down, their gradients are in closed form and can be computed very efficiently.

As is common in machine learning, since we know a computable formula of the Wasserstein distance, and we can take the gradient of that formula efficiently using analytic forms, we can optimize objective functions based on the Wasserstein distance using gradient-based optimization algorithms. For instance, given N functional connectomes $\boldsymbol{G}^{(1)}, \boldsymbol{G}^{(2)}, ..., \boldsymbol{G}^{(N)}$, we can determine a persistent diagram centroid $\overline{\boldsymbol{b}}$ (for the connected components) that minimizes the sum of the 2-Wasserstein distances as

$$\overline{\boldsymbol{b}} = \arg\min_{\boldsymbol{b}_{\overline{\boldsymbol{G}}}} \sum_{i=1}^{N} W_{2,B}(\overline{\boldsymbol{G}}, \boldsymbol{G}^{(i)}) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{b}_{\boldsymbol{G}^{(i)}}.$$

$\overline{\boldsymbol{b}}$ represents the Wasserstein barycenter that quantifies the central tendency of the functional connectomes. Likewise, the variability around the barycenter can be determined as

$$s_{\boldsymbol{b}}^2 = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{b}_{\boldsymbol{G}^{(i)}} - \overline{\boldsymbol{b}}).$$

$s_{\boldsymbol{b}}^2$ represents the Wasserstein variance. In a similar manner, the Wasserstein mean $\overline{\boldsymbol{d}}$ and variance $s_{\boldsymbol{d}}^2$ for the cycles can be calculated using $W_{p,D}$. Additionally, other important statistics that measure central tendency, dispersion, shape, and association can also be computed.

Scalability The computation of the Wasserstein distance is very efficient. By sorting birth and death values and matching them in order, the computational cost for evaluating $W_{p,B}$ and $W_{p,D}$ is $O(n \log n)$, where n is the number of edges in connectome graphs.

3 Connectome Analysis of Neural Networks

Drawing inspiration from functional MRI studies of the human brain, we are interested in the functional behavior of neural networks and whether we can characterize them using our proposed functional connectomes and persistent-graph-homology representation. Through two analysis studies, we aim to explore these connections.

In the first study, we will analyze how various popular regularization strategies, including batch normalization, dropout, and L^2 regularization, influence the overall functional mechanisms and data propagation in neural networks during inference. Regularization affects neural network weights similarly to how various factors influence structural brain networks obtained by diffusion MRI. This, in turn, impacts how functional mechanisms unfold.

In the second study, we will conduct a more fine-grained investigation into how neural networks process different stimuli through functional connectomes. Specifically, we will explore whether there are inherent topological patterns within the functional mechanisms of processing data points from various predefined classes. For example, we will analyze how neural networks process samples from different digit classes (0-9) in the MNIST dataset. This is analogous to how the human brain perceives and processes different visual stimuli, such as recognizing and distinguishing between digits, characterized by human connectomes observed in function MRI studies.

By drawing these comparisons, we aim to gain insights into the similarities between neural networks and human brain functions to better understand and characterize these systems.

Cluster analysis We are interested in identifying the natural groupings and relationships within the functional mechanism using our proposed persistent-graph-homology framework. Our goal is to determine if this framework can uncover the intrinsic structure of functional connectomes *without* supervision from predefined classes. Supervised learning can alter the original representation through coefficient and weight adjustments, potentially obscuring the underlying patterns and groupings. Clustering, however, can validate predefined classes by revealing if clusters of functional connectomes align well with them. Specifically, connectomes from the same class should be topologically similar and grouped into the same cluster, while connectomes from different classes should be dissimilar and placed in separate clusters. If this alignment occurs, it supports the idea that the functional mechanisms of neural networks are characterized by their topology and that our framework effectively captures these topological signals. Therefore, we will utilize unsupervised clustering to explore the data, grouping similar connectomes based on their subtle topological patterns.

Method comparison We evaluated the clustering performance of our proposed method, termed *Top*, relative to six other baseline methods. Our Top method utilizes centroid-based clustering that minimizes within-cluster variances based on squared 2-Wasserstein distances $W_{2,B}^2 + W_{2,D}^2$ of birth/death values and the Wasserstein barycenter, optimized via Lloyd's algorithm (Forgy 1965).

The first baseline method uses k-means clustering on the vectorization of entries below the main diagonal of adjacency (Adj) matrices of functional connectomes, grouping the connectomes based on node-by-node geometry. The remaining five methods use persistent homology and involve clustering on conventional Rips-complex persistence diagrams of the 1st homology group, commonly used in the literature. The k-medoids algorithm is applied using the following distances and kernels: bottleneck distance (BD), Wasserstein distance (WD), sliced Wasserstein kernel (SWK) (Carriere, Cuturi, and Oudot 2017), and heat kernel (HK) (Reininghaus et al. 2015). Additionally, k-means clustering is performed on the persistence image (PI) vectorization (Adams et al. 2017). More details on these methods are available in the extended version and code.

For all methods, initial clusters are selected at random, and we perform clustering 20 trials and report average clustering performance.

Datasets We performed our analyses on three datasets: MNIST (LeCun et al. 1998), Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), and CIFAR-10 (Krizhevsky, Hinton et al. 2009). MNIST comprises a collection of grayscale images of handwritten digits, Fashion-MNIST consists of grayscale images of fashion products, and CIFAR-10 includes color images of various animals and objects. Each of these datasets contains 10 predefined classes. More details on the datasets are available in the extended version.

Neural network architectures and training We employed neural network architectures of increasing complexity to match the varying complexities of the three datasets (MNIST, Fashion-MNIST, and CIFAR-10), while also keeping the architectures compact. This approach ensures that we can train well-generalized neural networks, which will then be analyzed for their behavior in our studies, while maintaining simplicity for transparency and interpretability of the analysis results. Additionally, conventional persistent homology methods, which will serve as baseline methods for comparison with our proposed method, have cubic time complexity (Otter et al. 2017), limiting the architecture size to fewer than a few hundred nodes, as demonstrated in the runtime experiment below.

For MNIST, we used a feedforward architecture with two hidden fully-connected layers, with the first and second layers comprising 128 and 64 neurons, respectively. For Fashion-MNIST, we used a similar feedforward architecture, but with the first and second layers comprising 256 and 128 neurons, respectively. For CIFAR-10, we used a convolutional neural network with three VGG blocks (Simonyan and Zisserman 2015), followed by two fully-connected layers, with the first and second layers comprising 256 and 128 neurons, respectively. In all architectures, we applied leaky ReLU activation functions and the stochastic gradient descent optimizer with momentum.

To train neural networks, we randomly partitioned the data points of each dataset into a training dataset and a functional dataset, as explained in Section 2. For each training strategy–namely 1) batch normalization, 2) dropout, 3) L^2, as well as 4) vanilla (which trains neural networks without any regularization and serves as a control)–we used the training set to optimize and obtain well-generalized neural networks. To account for the stochastic nature of gradient-based optimization initialization, we trained 20 neural networks for each strategy, totaling 80 networks (20 × 4). For the MNIST dataset, the average test accuracies are 0.98 across all training strategies. For Fashion-MNIST, the average test accuracies are 0.89 across all training strategies. For CIFAR-10, the average test accuracies are 0.76 across all training strategies.

More details on the architecture and hyperparameter tuning are available in the extended version and code.

Once the fully-trained neural networks are obtained, the functional dataset is fed into these networks to extract functional connectomes. Two different methods of extraction are employed for the two analysis studies, which will be provided in more detail below for each specific study. Note that only neurons in the hidden fully-connected layers are used to construct the connectomes; neurons in the softmax output layers and convolutional layers are excluded.

Study 1

We will analyze the influence of various popular regularization strategies on the *overall* functional mechanisms and data propagation in neural networks during inference. To construct functional connectomes for each dataset, we feed the *entire* corresponding functional dataset to the neural network to extract functional connectomes. As a result, we obtain 80 functional connectomes, with 20 of these connectomes from each training strategy (batch normalization, dropout, L^2, and vanilla). We will perform cluster analysis on these 80 data points to group them into four clusters. Figure 2 illustrates the average functional connectomes from training on the Fashion-MNIST dataset using each strategy, along with their corresponding persistence diagrams, which describe the topology. Additionally, we will cluster each regularization strategy against the control group (i.e., vanilla) to better understand the impact of each regularization method

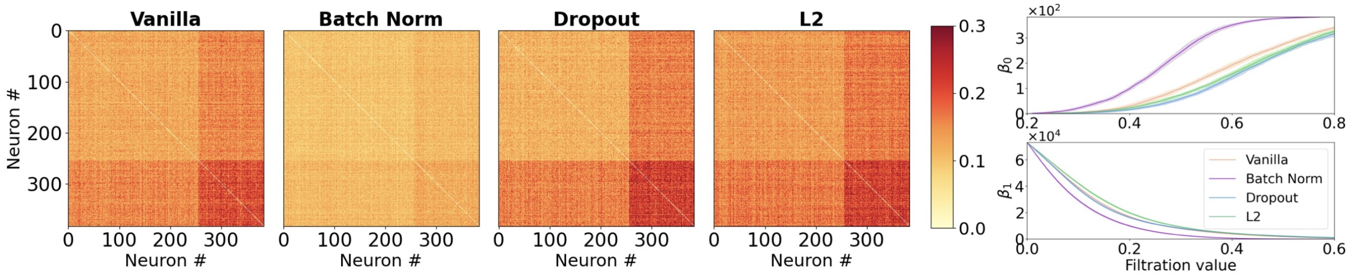

Figure 2: Statistics of the functional dataset used in Study 1. Left: Sample means of the functional connectomes, averaged within each training strategy. Right: Persistence diagrams and statistics for each strategy, with thick lines representing Wasserstein barycenters and shaded regions indicating Wasserstein standard deviation.

Dataset	Strategy	BD	WD	SWK	HK	PI	Adj	Top
MNIST	All	0.63 ± 0.04	0.75	**0.85**	0.72 ± 0.01	0.69 ± 0.01	0.32 ± 0.05	0.78 ± 0.11
	Batch Norm vs. Vanilla	**1.00**	**1.00**	**1.00**	0.93	0.93	0.55 ± 0.04	**1.00**
	Dropout vs. Vanilla	0.67 ± 0.07	0.90	0.95	0.98	0.98	0.54 ± 0.03	**1.00**
	L^2 vs. Vanilla	0.62 ± 0.04	0.70	0.70	0.60	0.58	0.55 ± 0.03	$\mathbf{0.80 \pm 0.03}$
Fashion-MNIST	All	0.68	0.75 ± 0.01	0.78	0.52 ± 0.01	0.53 ± 0.02	0.34 ± 0.07	$\mathbf{0.87 \pm 0.12}$
	Batch Norm vs. Vanilla	**1.00**	**1.00**	**1.00**	**1.00**	0.98	0.58 ± 0.07	**1.00**
	Dropout vs. Vanilla	0.73	0.95	0.93	0.50	0.50	0.54 ± 0.04	**0.98**
	L^2 vs. Vanilla	0.83	0.98	0.95	0.53	0.53	0.55 ± 0.03	**1.00**
CIFAR-10	All	0.75 ± 0.01	**0.98**	0.96	0.81	0.58 ± 0.02	0.51 ± 0.11	0.88 ± 0.11
	Batch Norm vs. Vanilla	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.56 ± 0.09	**1.00**
	Dropout vs. Vanilla	**1.00**	**1.00**	**1.00**	**1.00**	0.63 ± 0.07	0.62 ± 0.17	**1.00**
	L^2 vs. Vanilla	0.64 ± 0.01	**0.95**	0.93	0.68	0.67 ± 0.01	0.55 ± 0.04	0.94 ± 0.01

Table 1: Comparison of clustering performance in Study 1 across different datasets and training strategies, reported as average purity scores ± standard deviation. The table presents results for clustering *all* strategies together, as well as pairwise clustering analysis of each regularization strategy compared to the control group (vanilla).

on the neural networks' functional mechanisms. That is, we will cluster 40 data points into two groups.

Since these datasets are balanced, with 20 samples per class, we can evaluate clustering performance using *purity* (Manning, Raghavan, and Schütze 2008). The purity score ranges from 0 to 1, with 1 indicates perfect clustering alignment. This evaluation measure is not only transparent and interpretable but also effective for this study, where the number of clusters is small and the cluster sizes are balanced.

Table 1 presents the clustering performance comparison. The findings indicate that regularization strategies, which influence learnable weight adjustment of neural networks, significantly affect data propagation through functional mechanisms, similar to neuroanatomy in the human nervous system (Sporns 2016). Specifically, the clustering performance shows high purity scores, which suggest substantial differences between each regularization strategy vs. the control, highlighting the notable impact of these techniques. Furthermore, different regularization strategies lead to distinct functional mechanisms, resulting in high purity scores in cluster analysis for all four training strategies. Overall, topological signals, as measured by persistent homology methods, prove to be an effective means of characterizing neural network functions. In most settings, the proposed Top method outperforms other baselines.

Study 2

We will explore how fully-trained neural networks process different stimuli using functional connectomes. We construct these connectomes as follows. For each dataset, we partition the data into collections where each collection contains data points from a specific predefined class. For instance, in the MNIST dataset, we create 10 collections, each corresponding to a digit class (0-9). We then feed each collection into the fully-trained neural network to extract the functional connectomes for that particular class. For each training strategy, we obtain 20 functional connectomes per class, resulting in a total of 200 functional connectomes (20 × 10). We will perform cluster analysis on these 200 data points to group them into ten clusters.

As in Study 1, these datasets in Study 2 are balanced so we will also use the purity score to evaluate clustering performance.

Table 2 displays the clustering performance comparison. Topological methods, including WD, SWK and Top, effectively capture the functions distinct to each predefined class. They are the best performers that achieve purity scores between 0.5 and 0.6 in *unsupervised* settings, which are significantly better than the 0.1 score expected if clustering was made randomly. These findings show that samples from each class are processed through distinct functional mecha-

Dataset	Strategy	BD	WD	SWK	HK	PI	Adj	Top
MNIST	Vanilla	0.33	0.40	0.44 ± 0.02	0.36 ± 0.01	0.36 ± 0.01	0.21 ± 0.02	$\mathbf{0.47 \pm 0.02}$
	Batch Norm	0.36 ± 0.02	$\mathbf{0.50}$	0.48 ± 0.02	0.35	0.34 ± 0.01	0.22 ± 0.03	0.46 ± 0.02
	Dropout	0.33 ± 0.02	0.47	0.46 ± 0.01	0.34 ± 0.01	0.34 ± 0.01	0.18 ± 0.03	$\mathbf{0.57 \pm 0.02}$
	L^2	0.29 ± 0.01	0.44 ± 0.01	0.45 ± 0.01	0.34 ± 0.01	0.34 ± 0.01	0.19 ± 0.02	$\mathbf{0.48 \pm 0.02}$
Fashion-MNIST	Vanilla	0.39 ± 0.01	0.62 ± 0.01	$\mathbf{0.64 \pm 0.02}$	0.46 ± 0.02	0.43 ± 0.01	0.23 ± 0.03	0.53 ± 0.02
	Batch Norm	0.38 ± 0.02	0.58 ± 0.02	$\mathbf{0.60 \pm 0.01}$	0.40 ± 0.01	0.39 ± 0.01	0.20 ± 0.04	0.49 ± 0.02
	Dropout	0.41	$\mathbf{0.60}$	0.59 ± 0.04	0.45 ± 0.02	0.40 ± 0.01	0.19 ± 0.03	0.53 ± 0.03
	L^2	0.43	0.54 ± 0.01	$\mathbf{0.59 \pm 0.01}$	0.41	0.41 ± 0.01	0.21 ± 0.04	0.53 ± 0.04
CIFAR-10	Vanilla	0.30 ± 0.01	$\mathbf{0.56 \pm 0.01}$	0.55 ± 0.01	0.41 ± 0.01	0.40 ± 0.01	0.15 ± 0.02	0.51 ± 0.03
	Batch Norm	0.33 ± 0.01	0.51 ± 0.01	0.47	0.35 ± 0.01	0.35 ± 0.01	0.16 ± 0.01	$\mathbf{0.52 \pm 0.02}$
	Dropout	0.29 ± 0.01	0.49	$\mathbf{0.51 \pm 0.01}$	0.37 ± 0.01	0.37 ± 0.01	0.26 ± 0.04	0.50 ± 0.02
	L^2	0.35 ± 0.01	0.59 ± 0.03	$\mathbf{0.59 \pm 0.01}$	0.50 ± 0.01	0.48 ± 0.01	0.18 ± 0.03	0.54 ± 0.03

Table 2: Comparison of clustering performance in Study 2, reported as average purity scores ± standard deviation.

nisms. These phenomena are observed in all training strategies. This is similar to how the human brain uses specialized neural mechanisms to process different types of stimuli, ensuring efficient and effective interpretation of diverse information (Kandel et al. 2000).

Runtime Experiment

All topological methods used in the studies were evaluated through runtime experiments. These methods were executed on an Apple M1 Pro CPU with 16 GB of unified RAM. Figure 3 shows the plot of runtime vs. input size. The results clearly indicate that all five persistent homology-based distances and kernels (BD, WD, SWK, HK, and PI) are limited to handling dense graphs with only a few hundred nodes, highlighting the current scaling limitations of persistent homology embedding methods and their heavy reliance on approximation solutions. In contrast, Top can compute the exact Wasserstein distance between graphs with thousands of nodes and millions of edges in about one second. This computational efficiency makes Top practical for large-scale analyses of neural networks, which cannot be effectively analyzed using methods based on conventional persistence diagrams.

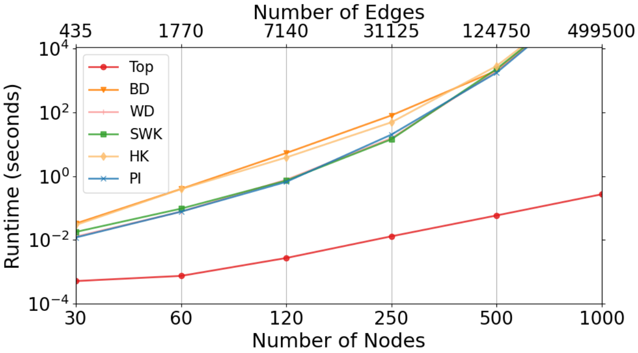

Figure 3: Average runtime of each method for computing topological distance or kernel between two complete graphs. The graphs were generated using a modular network approach (details in the extended version, with code provided). The runtime is plotted against the network size, represented by the number of nodes and edges.

Potential Impact Our approach for characterizing functional connectomes in neural networks is effective, computable, and scalable, potentially impacting the analysis of large neural architectures. By integrating neural network interpretability with human brain function insights, our framework opens up opportunities to leverage established techniques from functional MRI analysis. From a statistical learning perspective, our persistence statistics provide a robust basis for hypothesis testing and permutation tests, enhancing statistical rigor. Their linear-logarithmic efficiency supports large-scale neural network applications. Additionally, the gradient computability of the Wasserstein distance aids in designing advanced machine learning algorithms through gradient descent optimization.

Our studies on convolutional neural networks for the CIFAR-10 dataset demonstrate the effectiveness of focusing on subnetworks within the last few fully-connected layers, enabling topological analysis of more targeted functional mechanisms. This approach could be particularly effective for complex, deep neural networks, including those with multiple heads. While primarily focused on feedforward architectures, our method can also be extended to convolutional layers and recurrent networks.

Limitation Persistent graph homology is limited to the topological invariants of connected components and cycles. These two features, however, play a dominant role in topological analyses. For example, they are widely utilized in the brain network community (Bullmore and Sporns 2009; Honey et al. 2007), and cycles, in particular, have been increasingly reported as the most discriminative topological feature in brain networks (Sizemore et al. 2018), galaxy organization (Biagetti, Cole, and Shiu 2021), and protein structure (Xia and Wei 2014). In contrast, the assessment of higher-order features beyond cycles offers limited practical value due to their relative rarity, interpretive challenges, and consequent minimal discriminative power (Biagetti, Cole, and Shiu 2021; Sizemore et al. 2018; Songdechakraiwut and Chung 2020).

References

Adams, H.; Emerson, T.; Kirby, M.; Neville, R.; Peterson, C.; Shipman, P.; Chepushtanova, S.; Hanson, E.; Motta, F.; and Ziegelmeier, L. 2017. Persistence images: a stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8): 1–35.

Biagetti, M.; Cole, A.; and Shiu, G. 2021. The persistence of large scale structures. Part I. Primordial non-Gaussianity. *Journal of Cosmology and Astroparticle Physics*, 2021(04): 061.

Bullmore, E.; and Sporns, O. 2009. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3): 186–198.

Carriere, M.; Cuturi, M.; and Oudot, S. 2017. Sliced Wasserstein kernel for persistence diagrams. In *International Conference on Machine Learning (ICML)*, 664–673.

Edelsbrunner, H.; and Harer, J. L. 2022. *Computational Topology: An Introduction*. American Mathematical Society.

Forgy, E. W. 1965. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21: 768–769.

Fornito, A.; Zalesky, A.; and Bullmore, E. 2016. *Fundamentals of Brain Network Analysis*. Academic press.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Honey, C. J.; Kötter, R.; Breakspear, M.; and Sporns, O. 2007. Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *Proceedings of the National Academy of Sciences*, 104(24): 10240–10245.

Honey, C. J.; Sporns, O.; Cammoun, L.; Gigandet, X.; Thiran, J.-P.; Meuli, R.; and Hagmann, P. 2009. Predicting human resting-state functional connectivity from structural connectivity. *Proceedings of the National Academy of Sciences*, 106(6): 2035–2040.

Kandel, E. R.; Schwartz, J. H.; Jessell, T. M.; Siegelbaum, S.; Hudspeth, A. J.; Mack, S.; et al. 2000. *Principles of Neural Science*, volume 4. McGraw-hill New York.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Kwon, Y.-K.; and Cho, K.-H. 2007. Analysis of feedback loops and robustness in network evolution based on Boolean models. *BMC Bioinformatics*, 8.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lee, H.; Kang, H.; Chung, M. K.; Kim, B.-N.; and Lee, D. S. 2012. Persistent brain network homology from the perspective of dendrogram. *IEEE Transactions on Medical Imaging*, 31(12): 2267–2277.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Naitzat, G.; Zhitnikov, A.; and Lim, L.-H. 2020. Topology of deep neural networks. *Journal of Machine Learning Research*, 21(184): 1–40.

Otter, N.; Porter, M. A.; Tillmann, U.; Grindrod, P.; and Harrington, H. A. 2017. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6: 1–38.

Reininghaus, J.; Huber, S.; Bauer, U.; and Kwitt, R. 2015. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4741–4748.

Rieck, B.; Togninalli, M.; Bock, C.; Moor, M.; Horn, M.; Gumbsch, T.; and Borgwardt, K. 2019. Neural persistence: a complexity measure for deep neural networks using algebraic topology. In *International Conference on Learning Representations (ICLR)*.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*.

Sizemore, A. E.; Giusti, C.; Kahn, A.; Vettel, J. M.; Betzel, R. F.; and Bassett, D. S. 2018. Cliques and cavities in the human connectome. *Journal of Computational Neuroscience*, 44: 115–145.

Skraba, P.; and Turner, K. 2023. Wasserstein stability for persistence diagrams. *arXiv preprint arXiv:2006.16824*.

Songdechakraiwut, T.; and Chung, M. K. 2020. Dynamic topological data analysis for functional brain signals. In *IEEE International Symposium on Biomedical Imaging*, 1–4.

Songdechakraiwut, T.; and Chung, M. K. 2023. Topological learning for brain networks. *The Annals of Applied Statistics*, 17(1): 403.

Songdechakraiwut, T.; Krause, B. M.; Banks, M. I.; Nourski, K. V.; and Van Veen, B. D. 2023. Wasserstein distance-preserving vector space of persistent homology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 277–286.

Songdechakraiwut, T.; Krause, B. M.; Banks, M. I.; Nourski, K. V.; and Veen, B. D. V. 2022. Fast topological clustering with Wasserstein distance. In *International Conference on Learning Representations (ICLR)*.

Songdechakraiwut, T.; Shen, L.; and Chung, M. 2021. Topological learning and its application to multimodal brain network integration. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 166–176.

Sporns, O. 2016. *Networks of the Brain*. MIT press.

Tewarie, P.; van Dellen, E.; Hillebrand, A.; and Stam, C. J. 2015. The minimum spanning tree: an unbiased method for brain network analysis. *NeuroImage*, 104: 177–188.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Watanabe, S.; and Yamana, H. 2022. Topological measurement of deep neural networks using persistent homology.

Annals of Mathematics and Artificial Intelligence, 90(1): 75–92.

Xia, K.; and Wei, G.-W. 2014. Persistent homology analysis of protein structure, flexibility, and folding. *International Journal for Numerical Methods in Biomedical Engineering*, 30(8): 814–844.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Zhang, B.; Dong, Z.; Zhang, J.; and Lin, H. 2023. Functional network: a novel framework for interpretability of deep neural networks. *Neurocomputing*, 519: 94–103.