DeepFDR: A Deep Learning-based False Discovery Rate Control Method for Neuroimaging Data

Taehyo Kim[†,1] **Hai Shu**[†,‡,1] **Qiran Jia**[1,2] **Mony J. de Leon**[3]
for the Alzheimer's Disease Neuroimaging Initiative*
[1]Department of Biostatistics, School of Global Public Health, New York University
[2]Department of Population and Public Health Sciences, University of Southern California
[3]Brain Health Imaging Institute, Department of Radiology, Weill Cornell Medicine
[†]Equal contributions
[‡]Correspondence: hs120@nyu.edu

Abstract

Voxel-based multiple testing is widely used in neuroimaging data analysis. Traditional false discovery rate (FDR) control methods often ignore the spatial dependence among the voxel-based tests and thus suffer from substantial loss of testing power. While recent spatial FDR control methods have emerged, their validity and optimality remain questionable when handling the complex spatial dependencies of the brain. Concurrently, deep learning methods have revolutionized image segmentation, a task closely related to voxel-based multiple testing. In this paper, we propose DeepFDR, a novel spatial FDR control method that leverages unsupervised deep learning-based image segmentation to address the voxel-based multiple testing problem. Numerical studies, including comprehensive simulations and Alzheimer's disease FDG-PET image analysis, demonstrate DeepFDR's superiority over existing methods. DeepFDR not only excels in FDR control and effectively diminishes the false nondiscovery rate, but also boasts exceptional computational efficiency highly suited for tackling large-scale neuroimaging data.

1 INTRODUCTION

Voxel-based multiple testing is widely used in neuroimaging data analysis (Ashburner and Friston, 2000; Genovese et al., 2002; Mirman et al., 2018). For instance, in Alzheimer's disease research, as a neurodegeneration biomarker, Fluorine-18 fluorodeoxyglucose positron emission tomography (FDG-PET) measures the brain glucose metabolism and is extensively used for early diagnosis and monitoring the progression of Alzheimer's disease (Alexander et al., 2002; Drzezga et al., 2003; Shivamurthy et al., 2015; Ou et al., 2019). To statistically compare brain glucose metabolism between two groups of different disease statuses, FDG-PET studies in Alzheimer's disease often conduct multiple testing at the voxel level to identify brain regions with functional abnormalities (Mosconi et al., 2005; Lee et al., 2015; Shu et al., 2015; Kantarci et al., 2021).

The prevalent multiple testing methods are based on controlling the *false discovery rate* (FDR; Benjamini and Hochberg (1995)), an alternative yet more powerful measure of type I error than the conventional family-wise error rate (FWER). The corresponding measure of type II error is the *false nondiscovery rate* (FNR; Genovese and Wasserman (2002)). However, for neuroimaging data, traditional FDR control methods such as the BH (Benjamini and Hochberg, 1995), q-value (Storey et al., 2003), and LocalFDR (Efron, 2004) methods, ignore the spatial dependence among the voxel-based tests and thus suffer from substantial loss of testing power (Shu et al., 2015). The voxel-based tests are inherently dependent due to the spatial structure among brain voxels. Although some FDR control methods, applicable to spatial and three-dimensional (3D) contexts, recently have been developed, they either use basic spatial models such as simple hidden Markov random fields (HMRF; Shu et al. (2015); Liu et al. (2016); Kim et al. (2018)) and simple Gaussian random fields (Sun et al., 2015), or rely on local smoothing approaches (Tansey et al., 2018a; Cai et al., 2022; Han et al., 2023). The *validity* of these methods in controlling FDR and their *op-*

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

timality in minimizing FNR are called into question when handling the imaging data of the complex human brain, which is spatially heterogeneous due to its anatomical structure (Brodmann, 2007) and exhibits long-distance functional connectivity between brain regions (Liu et al., 2013). Hence, it is imperative to develop a spatial FDR control method that effectively captures the brain's intricate dependencies and enjoys theoretical guarantees of the validity and optimality.

It is noteworthy that the aforementioned methods of Shu et al. (2015), Liu et al. (2016), Kim et al. (2018) and Sun et al. (2015) all use a testing procedure introduced by Sun and Cai (2009), which relies on the local index of significance (LIS) rather than the more commonly used p-value. Unlike the p-value, which is determined solely by the test statistic at the corresponding spatial location, the LIS at any spatial location is the conditional probability that its null hypothesis is true, given the test statistics from all spatial locations. Under mild conditions, the LIS-based testing procedure can asymptotically minimize the FNR while controlling the FDR under a prespecified level (Sun and Cai, 2009; Xie et al., 2011). Thus, the performance of the LIS-based testing procedure hinges on the capability of the selected spatial model to appropriately model the dependencies among the tests.

A task closely related to voxel-based multiple testing is image segmentation (Minaee et al., 2021). Both follow a procedure where the input is an image: a map of test statistics for multiple testing, and the target image for segmentation; the output assigns a label to each voxel/pixel: hypothesis state labels in multiple testing, and segmentation labels in image segmentation. This similarity prompts the question: can we apply image segmentation models to voxel-based multiple testing?

In medical image segmentation, deep learning methods, especially the U-net and its variants (Ronneberger et al., 2015; Çiçek et al., 2016; Isensee et al., 2021; Chen et al., 2021; Cao et al., 2022b; Hatamizadeh et al., 2022; Pan et al., 2023), have established state-of-the-art results. The foundational U-net architecture consists of a contracting path designed to extract the global salient features and an expanding path utilized to recover local spatial details through skip connections from the contracting path. This innovative network design empowers these network models to effectively capture both short and long-range spatial dependencies and account for spatial heterogeneity. The U-net and its variants have emerged as top performers in various segmentation tasks for neuroimaging data. These include challenges like the Brain Tumor Segmentation (BraTS) Challenge (Bakas et al., 2019), the Ischemic Stroke Lesion Segmentation (ISLES) Challenge (Liew et al., 2022), and the Infant Brain MRI Segmentation (iSeg) Challenge (Sun et al., 2021).

However, our voxel-based multiple testing is an unsupervised learning task without ground-truth hypothesis state labels, contrasting with most deep-learning methods for image segmentation, which are supervised and require predefined ground-truth labels during training (Siddique et al., 2021). Recently, several unsupervised deep learning-based image segmentation methods have been developed. Xia and Kulis (2017) proposed the W-net, a cascade of two U-nets, where the normalized cut loss of the first U-net and the reconstruction loss of the second U-net are iteratively minimized to generate segmentation probability maps. Kanezaki (2018) utilized a convolutional neural network (CNN) to extract features, clustered them for pseudo labels, and alternately optimized the pseudo labels and segmentation network through self-training. Kim et al. (2020) further improved upon this approach by introducing a spatial continuity loss. Pu et al. (2023) designed an autoencoder network integrated with an expectation-maximization module, which employs a Gaussian mixture model to relate segmentation labels to the deep features extracted from the encoder and constrained by image reconstruction via the decoder, and ultimately assigns labels based on their conditional probabilities given these deep features.

In this paper, we propose DeepFDR, a novel deep learning-based FDR control method for voxel-based multiple testing. We innovatively connect the voxel-based multiple testing with the deep learning-based unsupervised image segmentation. Specifically, we adopt the LIS-based testing procedure (Sun and Cai, 2009), where the LIS values are estimated by the segmentation probability maps from our modified version of the W-net (Xia and Kulis, 2017). The aforementioned unsupervised image segmentation methods of Kanezaki (2018), Kim et al. (2020) and Pu et al. (2023) are not applicable in this context, as they do not estimate the conditional probability of each voxel's label given the input image, which coincides with the LIS when the input is the map of test statistics.

To the best of our knowledge, our work is the first to directly apply deep learning to unsupervised spatial multiple testing. We notice that four recent studies (Xia et al., 2017; Tansey et al., 2018b; Romano et al., 2020; Marandon et al., 2022) have also used deep neural networks in multiple testing, but there are intrinsic distinctions between their approaches and ours. Xia et al. (2017) proposed the NeuralFDR method to address multiple testing problems when covariate information for each hypothesis test is available. NeuralFDR employs a deep neural network to learn the p-value threshold as a function of the covariates. Although 3D coordinates may serve as covariates, Neu-

ralFDR assumes that the p-value and covariates for each test are independent under the null hypothesis but dependent under the alternative. This assumption does not align with the nature of spatial data, where true and false nulls can be spatially adjacent. Tansey et al. (2018b) developed the BB-FDR method for independent tests each with covariates, in contrast to the dependent tests in our study. BB-FDR uses a deep neural network to model the hyperprior parameters of the hypothesis state based on the covariates. Romano et al. (2020) introduced Deep Knockoffs, a method that employs a deep neural network to generate model-X knockoffs, but their model-X knockoffs problem is different from our voxel-based multiple testing problem. Marandon et al. (2022) applied neural networks as classifiers to solve a semi-supervised multiple testing problem, where a subset of the sample data, termed a null training sample (NTS), is known from the null distribution. Their method is not applicable to our unsupervised voxel-based multiple testing due to the absence of an NTS. In our context, even if an NTS might be additionally generated from a known null distribution, it would not offer useful spatial dependence information.

Our contributions are summarized as follows:

- We propose DeepFDR, a pioneering method that harmoniously combines deep learning techniques with voxel-based multiple testing. Inspired by advancements in unsupervised image segmentation, DeepFDR offers a fresh perspective on controlling the FDR in neuroimaging analyses.
- We empirically demonstrate the superior performance of DeepFDR through rigorous simulation studies and in-depth analysis of 3D FDG-PET images pertaining to Alzheimer's disease. Our findings indicate its consistent capability to adeptly control FDR whilst effectively reducing FNR, thereby ensuring enhanced reliability of results.
- DeepFDR exhibits exceptional computational efficiency by leveraging the mature software and advanced optimization algorithms from deep learning. This advantage distinguishes it from existing spatial FDR control methods, rendering it highly suited for handling large-scale neuroimaging data.

A Python package for our DeepFDR method is available at `https://github.com/kimtae55/DeepFDR`.

2 METHOD

2.1 Problem Formulation

Consider two population groups, for example, the Alzheimer's disease group and the cognitively normal group. We aim to compare the brain glucose metabolism between the two groups by testing the difference in their voxel-level population means of the standardized uptake value ratio (SUVR) from FDG-PET. Each subject in the sample data has a 3D brain FDG-PET image with m voxels of interest. Let x_i be a test statistic for the null hypothesis $\mathcal{H}_i$, which assumes that there is no difference in the mean values of SUVR between the two groups at voxel i. The unobservable state label h_i is defined as $h_i = 1$ if $\mathcal{H}_i$ is false and $h_i = 0$ otherwise. The goal of multiple testing is to predict the unknown labels $\boldsymbol{h} = [h_1, ..., h_m]$ based on the test statistics $\boldsymbol{x} = [x_1, ..., x_m]$. Table 1 summarizes the classification of tested hypotheses. The FDR and FNR are defined as

$$FDR = E\left[\frac{N_{10}}{R \vee 1}\right] \quad \text{and} \quad FNR = E\left[\frac{N_{01}}{A \vee 1}\right], \quad (1)$$

where $a \vee b = \max(a, b)$. An FDR control method is *valid* if it controls FDR at a prespecified level, and is *optimal* if it has the smallest FNR among all valid FDR control methods. We aim to develop an optimal FDR control method for voxel-based multiple testing. For simplicity, false nulls and rejected nulls are called *signals* and *discoveries*, respectively.

Number	Not rejected	Rejected	Total
True null	N_{00}	N_{10}	m_0
False null	N_{01}	N_{11}	m_1
Total	A	R	m

Table 1: Classification of tested hypotheses

2.2 LIS-based Testing Procedure

Sun and Cai (2009) defined the LIS for hypothesis $\mathcal{H}_i$ by

$$LIS_i(\boldsymbol{x}) = P(h_i = 0|\boldsymbol{x}), \quad (2)$$

which depends on all test statistics $\boldsymbol{x} = [x_1, ..., x_m]$, not just the local statistic x_i. They proposed the LIS-based testing procedure for controlling FDR at a prespecified level α:

$$\begin{aligned} &\text{Let } k = \max\{j : \frac{1}{j}\sum_{i=1}^{j} LIS_{(i)}(\boldsymbol{x}) \le \alpha\}, \\ &\text{then reject all } \mathcal{H}_{(i)} \text{ with } i = 1, ..., k. \end{aligned} \quad (3)$$

Here, $LIS_{(1)}(\boldsymbol{x}), \ldots, LIS_{(m)}(\boldsymbol{x})$ are the ranked LIS values in ascending order and $\mathcal{H}_{(1)}, \ldots, \mathcal{H}_{(m)}$ are the corresponding null hypotheses. In this procedure, $\{LIS_i(\boldsymbol{x})\}_{i=1}^m$ are practically replaced with their estimates, denoted by $\{\widehat{LIS}_i(\boldsymbol{x})\}_{i=1}^m$. Due to the identity

$$FDR = E\left[\frac{1}{R \vee 1}\sum_{i=1}^{R} LIS_{(i)}(\boldsymbol{x})\right],$$

the LIS-based testing procedure in (3) is valid for controlling FDR at level α. Under mild conditions, this procedure is asymptotically optimal in minimizing the FNR (Sun and Cai, 2009; Xie et al., 2011; Shu et al.,

2015). The LIS theory of Sun and Cai (2009) is applicable to spatial models that satisfy a monotone ratio condition (MRC) (their equation (3)). While their article primarily illustrates the theory through hidden Markov models (HMM), it also acknowledges the broad applicability of the MRC. The theory is extendable to a generalized MRC in Shu et al. (2015) (their equation (B.1)). Thus, one needs the generalized MRC rather than HMM to apply the LIS theory.

2.3 DeepFDR

Most deep learning-based methods for image segmentation produce segmentation probability maps $\{\{\widehat{P}(s_i = k|\boldsymbol{x})\}_{i=1}^m\}_{k=0}^{K-1}$ as the basis for label assignment, where $\boldsymbol{x}$ is the input image for segmentation, s_i is the segmentation label at the i-th voxel/pixel, and K is the number of label classes. We establish a connection between the image segmentation with $K = 2$ classes and voxel-based multiple testing by letting the input image for segmentation $\boldsymbol{x}$ be the 3D map of test statistics and assuming that segmentation label $s_i = k$ corresponds to the null hypothesis state $h_i = k$ for $k = 0, 1$. Consequently, the segmentation probability map $\{\widehat{P}(s_i = 0|\boldsymbol{x})\}_{i=1}^m$ may serve as an estimate of the LIS map $\{LIS_i(\boldsymbol{x}) = P(h_i = 0|\boldsymbol{x})\}_{i=1}^m$. This insight motivates us to adopt a deep learning-based image segmentation method for voxel-based multiple testing. As mentioned in Section 1, only unsupervised image segmentation methods are potentially suitable for our multiple testing problem. Particularly, the W-net (Xia and Kulis, 2017) is unsupervised and also generates the segmentation probability map. Moreover, the U-net structure used by the W-net excels at capturing multi-scale spatial information, effectively addressing short and long-range spatial dependencies as well as spatial heterogeneity. Thus, we choose to adopt the W-net and make slight modifications for multiple testing purposes. We then use its segmentation probability map as an estimate of the LIS map for the LIS-based testing procedure given in (3).

Figure 1 provides an overview of our DeepFDR architecture, which is based on the W-net. The input data for the network include the 3D map of test statistics $\boldsymbol{x} = [x_1, \ldots, x_m]$ and its corresponding 3D map of p-values $\boldsymbol{p} = [p_1, \ldots, p_m]$. The network consists of two cascaded U-nets. The first U-Net, $\boldsymbol{U}_1$, generates the segmentation probability map $\{\widehat{P}(s_i = 0|\boldsymbol{x})\}_{i=1}^m$ using the soft normalized cut (Ncut) loss given in (4). The second U-Net, $\boldsymbol{U}_2$, reconstructs the p-values $\boldsymbol{p}$ from the segmentation probability map using the mean squared error in (5) as the reconstruction loss. The soft Ncut loss plays a crucial role in partitioning the test statistics $\boldsymbol{x}$ into meaningful clusters, akin to the segmentation of an image. The reconstruction loss refines the segmentation probability map by enforcing the map to retain sufficient information from the input image. The two loss functions are alternately minimized, following the algorithm outlined in Algorithm 1. This iterative process results in the final segmentation probability map $\{\widehat{P}(s_i = 0|\boldsymbol{x})\}_{i=1}^m$. Subsequently, this map is fed into our LIS module to obtain the estimated LIS map $\{\widehat{LIS}_i(\boldsymbol{x})\}_{i=1}^m$ as per (6). Finally, this LIS map is plugged into the LIS-based testing procedure (3) to yield the multiple testing results. DeepFDR combines the strengths of deep learning-based image segmentation with the LIS-based testing procedure to effectively handle voxel-based multiple testing tasks. The key components of the network are elaborated below.

Soft Ncut loss. We use the soft Ncut loss as the loss function for the first U-net $\boldsymbol{U}_1$. The original Ncut loss (Shi and Malik, 2000) is widely used in data clustering and image segmentation. The loss for two classes is

$$Ncut_2(V) = \sum_{k=0}^{1} \frac{cut(A_k, V \setminus A_k)}{assoc(A_k, V)}$$
$$= 2 - \sum_{k=0}^{1} \frac{assoc(A_k, A_k)}{assoc(A_k, V)} = 2 - \sum_{k=0}^{1} \frac{\sum_{i\in A_k, j\in A_k} w_{ij}}{\sum_{i\in A_k, j\in V} w_{ij}},$$

where V is the set of all voxels, A_k is the set of voxels in class k, $cut(A, V \setminus A) = \sum_{i\in A, j\in V\setminus A} w_{ij}$ is the total weight of the edges that can be removed between sets A and $V \setminus A$, and $assoc(A, B) = \sum_{i\in A, j\in B} w_{ij}$ is the total weight of edges connecting voxels in set A to all voxels in set B. Minimizing the Ncut loss can simultaneously minimize the total normalized disassociation between classes and maximize the total normalized association within classes. To obtain the sets A_0 and A_1, the argmax function is used to assign the label $k_i^* = \arg\max_{k\in\{0,1\}} \widehat{P}(s_i = k|\boldsymbol{x})$ to each i-th voxel. To avoid the nondifferetiable argmax function in computing the Ncut loss, Xia and Kulis (2017) proposed the soft Ncut loss, which is differentiable, by using the soft labels $\{\{\widehat{P}(s_i = k|\boldsymbol{x})\}_{i=1}^m\}_{k=0}^1$ instead of the hard labels $\{k_i^*\}_{i=1}^m$. This allows the loss to be minimized using gradient descent algorithms for the W-net. The soft Ncut loss for two classes is defined as

$$L_{\text{soft-Ncut}}(\boldsymbol{\theta}_1) \tag{4}$$
$$= 2 - \sum_{k=0}^{1} \frac{\sum_{1\le i,j\le m} w_{ij}\widehat{P}(s_i = k|\boldsymbol{x})\widehat{P}(s_j = k|\boldsymbol{x})}{\sum_{1\le i,j\le m} w_{ij}\widehat{P}(s_i = k|\boldsymbol{x})},$$

where

$$[\widehat{P}(s_i = 0|\boldsymbol{x})]_{i=1}^m = [1 - \widehat{P}(s_i = 1|\boldsymbol{x})]_{i=1}^m = \boldsymbol{U}_1(\boldsymbol{x}; \boldsymbol{\theta}_1)$$

is the segmentation probability map obtained from the first U-net $\boldsymbol{U}_1$ with parameters $\boldsymbol{\theta}_1$, the weight

$$w_{ij} = \exp\left(-\frac{|x_i - x_j|^2}{\sigma_x^2} - \frac{\|\boldsymbol{\ell}_i - \boldsymbol{\ell}_j\|_2^2}{\sigma_\ell^2}\right) I(\|\boldsymbol{\ell}_i - \boldsymbol{\ell}_j\|_\infty \le r),$$

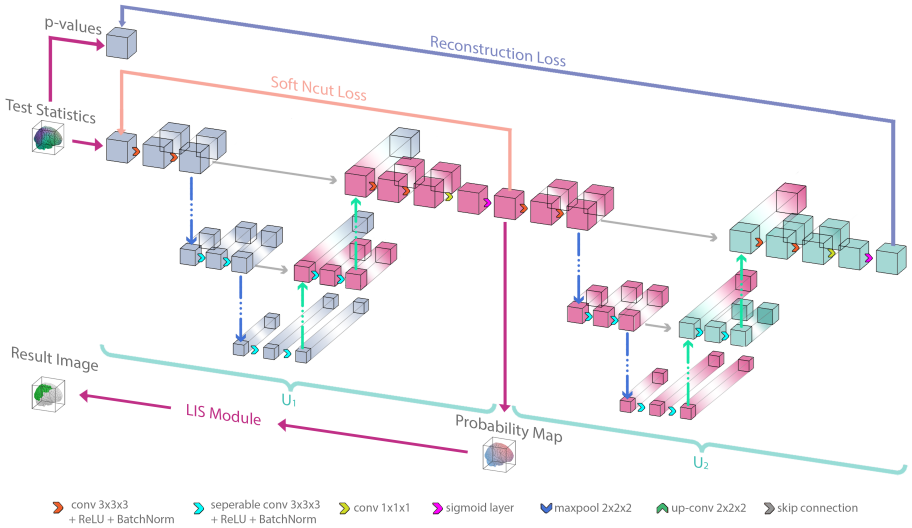

Figure 1: The network architecture of DeepFDR.

with $(\sigma_x, \sigma_\ell, r) = (11, 3, 3)$ in our paper, $\boldsymbol{\ell}_i$ contains the 3D coordinates, and $I(\cdot)$ is the indicator function.

Reconstruction loss. We use the mean squared error as the reconstruction loss for the second U-net $\boldsymbol{U}_2$:

$$L_{\text{recon}}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \frac{1}{m}\sum_{i=1}^{m}(p_i - \widehat{p}_i)^2 \tag{5}$$

where $\widehat{\boldsymbol{p}} = [\widehat{p}_i]_{i=1}^m = \boldsymbol{U}_2([\widehat{P}(s_j = 0|\boldsymbol{x})]_{j=1}^m; \boldsymbol{\theta}_2)$
$= \boldsymbol{U}_2(\boldsymbol{U}_1(\boldsymbol{x}; \boldsymbol{\theta}_1); \boldsymbol{\theta}_2)$

are the reconstructed p-values obtained from the second U-net $\boldsymbol{U}_2$ with parameters $\boldsymbol{\theta}_2$. Unlike the original W-net, we use the p-values $\boldsymbol{p}$ for reconstruction rather than the target image $\boldsymbol{x}$, which is the map of test statistics in our context. This modification is made because the reconstructed p-values $\widehat{\boldsymbol{p}}$ can be effectively constrained within the range [0,1] using a sigmoid layer. In contrast, if we were to use the reconstructed test statistics $\widehat{\boldsymbol{x}}$, they might not have a well-defined range if the original $\boldsymbol{x}$ (e.g., t-statistics) lacks one. Our initial simulation study also indicated that using p-values for reconstruction yields superior results. Parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are simultaneously updated in the minimization of the reconstruction loss.

LIS module and label flipping. The LIS module is a novel addition to the W-net architecture, enabling the implementation of the LIS-based testing procedure (3). Note that the final segmentation probability map $\{\widehat{P}(s_i = 0|\boldsymbol{x})\}_{i=1}^m$ from $\boldsymbol{U}_1$ cannot be directly used as the estimated LIS map $\{\widehat{LIS}_i(\boldsymbol{x})\}_{i=1}^m$. Since the segmentation process here is unsupervised without ground-truth labels, the segmentation label classes may be arbitrarily encoded as "0" and "1", potentially not corresponding well to the hypothesis state label classes. For example, it is possible that segmentation label $s_i = 1$ ($s_i = 0$) corresponds to hypothesis state label $h_i = 0$ ($h_i = 1$, resp.). To address this issue, we perform label flipping to correct the possible discrepancy. We compare the sets of significant voxels discovered by the LIS-based testing procedure based on $\widehat{LIS}_i(\boldsymbol{x}) = \widehat{P}(s_i = 0|\boldsymbol{x})$ and $\widehat{LIS}_i(\boldsymbol{x}) = \widehat{P}(s_i = 1|\boldsymbol{x})$, respectively, denoted as $S_{\widehat{P}_0}(\alpha)$ and $S_{\widehat{P}_1}(\alpha)$, with the discovery set $S_Q(\alpha)$ obtained using the q-value method. Since approximately $100(1-\alpha)\%$ of voxels in $S_Q(\alpha)$ are true signals due to the robust FDR control of the q-value method, our DeepFDR's discovery set is expected to encompass the majority of voxels in $S_Q(\alpha)$. Here, we use $S_Q(\alpha)$ as the reference set, owing to the q-value method's superior performance over BH and LocalFDR methods and its faster computation than other spatial FDR methods as shown in our simulation. We apply the widely-used Dice similarity coefficient (Dice, 1945) to measure the similarity between $S_{\widehat{P}_0}(\alpha)$ or $S_{\widehat{P}_1}(\alpha)$ and $S_Q(\alpha)$. The Dice coefficient for any two sets A and B is defined as the normalized size of their intersection:

$$Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|}.$$

If $Dice(S_{\widehat{P}_0}(\alpha), S_Q(\alpha)) < Dice(S_{\widehat{P}_1}(\alpha), S_Q(\alpha))$, we flip the segmentation label classes. Equivalently, the label flipping is performed as follows:

$$\widehat{LIS}_i(\boldsymbol{x}) \overset{\text{def}}{=} \widehat{P}(h_i = 0|\boldsymbol{x}) \tag{6}$$
$$= \begin{cases} \widehat{P}(s_i = 0|\boldsymbol{x}), & \text{if } Dice(S_{\widehat{P}_0}(\alpha), S_Q(\alpha)) \\ & \quad \geq Dice(S_{\widehat{P}_1}(\alpha), S_Q(\alpha)); \\ \widehat{P}(s_i = 1|\boldsymbol{x}), & \text{otherwise.} \end{cases}$$

If the q-value method yields no or a very small number of discoveries, one may gradually increase the nominal FDR level $\alpha_Q \geq \alpha$ exclusively for the q-value method

to obtain an acceptable $S_Q(\alpha_Q)$, and then apply the criterion (6). If $|S_Q(\alpha_Q)|$ remains very small despite a significant increase in α_Q compared to the original α, one may consider using p-values instead. For example, gradually decrease the uncorrected significance level $\alpha_P \leq \alpha$ for p-values, and in (6) replace $S_Q(\alpha_Q)$ with $S_P(\alpha_P)$, which is the set of voxels with p-values $< \alpha_P$. It is important to assume that the uncorrected p-value rejection set $S_P(\alpha)$ at level α is not excessively small; otherwise, one may need to contemplate increasing the nominal FDR level α for the multiple testing problem.

Algorithm 1 Algorithm for DeepFDR

Input: 3D volumes of test statistics $\boldsymbol{x}$ and p-values $\boldsymbol{p}$, and prespecified FDR level α.

1: **for** epoch $t = 1 : T$ **do**
2: Only update parameter $\boldsymbol{\theta}_1$ by minimizing the $L_{\text{soft-Ncut}}$ in (4);
3: Update both parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ by minimizing the L_{recon} in (5);
4: **end for**
5: Compute the LIS estimates $\{\widehat{LIS}_i(\boldsymbol{x})\}_{i=1}^m$ by (6);
6: Conduct the LIS-based testing procedure (3) with $\{\widehat{LIS}_i(\boldsymbol{x})\}_{i=1}^m$;

Output: A 3D volume of estimates for the null hypothesis states $\boldsymbol{h}$.

Detailed network architecture. Our DeepFDR network architecture, as depicted in Figure 1, is primarily based on the structure of the W-net (Xia and Kulis, 2017). It comprises two cascaded U-nets, each featuring a contracting path and an expanding path that span three levels of network layers. The network is equipped with a total of 10 pairs of two consecutive $3 \times 3 \times 3$ convolution layers, which have 64, 128, and 256 feature channels at the top, middle, and bottom levels, respectively. Each of these convolution layers is followed by a rectified linear unit (ReLU; Nair and Hinton (2010)) and batch normalization (Ioffe and Szegedy, 2015). While regular convolutions are utilized at the top level, depthwise separable convolutions (Chollet, 2017) are employed at the other two levels to significantly reduce parameters. The feature maps are downsampled from upper levels to lower levels by a $2 \times 2 \times 2$ max-pooling operation with a stride of 2 to halve spatial dimensions, but they are upsampled from lower levels to upper levels by a $2 \times 2 \times 2$ transposed convolution with a stride of 2 to double spatial dimensions. Skip connections are used to concatenate the feature maps in the contracting path with those in the expanding path to capture the multi-scale spatial information. Within each U-net, the last two layers consist of a $1 \times 1 \times 1$ convolution layer and a sigmoid layer. The convolution layer transforms all feature maps into a single feature map, enabling the subsequent sigmoid layer to generate the segmentation probability map $\{\widehat{P}(s_i = 0|\boldsymbol{x})\}_{i=1}^m$ for $\boldsymbol{U}_1$ or the reconstructed p-value map $\widehat{\boldsymbol{p}}$ for $\boldsymbol{U}_2$. The segmentation probability map $\{\widehat{P}(s_i = 0|\boldsymbol{x})\}_{i=1}^m$ from $\boldsymbol{U}_1$ and the input test statistics $\boldsymbol{x}$ are used to minimize the soft Ncut loss given in (4) with parameter $\boldsymbol{\theta}_1$, and the reconstructed and original p-value maps $\widehat{\boldsymbol{p}}$ and $\boldsymbol{p}$ are used to minimize the reconstruction loss given in (5) with parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$.

Network training. In contrast to supervised deep learning models which have access to multiple images with predefined ground-truth labels for training and validation, voxel-based multiple testing, as an unsupervised-learning problem, only has a single image of the test statistics $\boldsymbol{x}$ and thus has no straightforward validation set, and moreover lacks very effective validation criteria due to the absence of predefined ground-truth labels. While one might consider splitting the image of $\boldsymbol{x}$ into patches, this approach would lose long-range spatial structures and ignore the spatial heterogeneity. Alternatively, one could divide the sample data (e.g., subjects' FDG-PET images) into two parts and compute their respective maps of test statistics for training and validation, but the reduced sample size leads to less powerful test statistics. In our method, we utilize the complete map of test statistics from all sample data as the training image, and do not allocate an image for validation according to the W-net paper (Xia and Kulis, 2017). Instead, multiple regularization techniques are applied to prevent overfitting (Buhmann and Held, 1999) and enhance training stability: a dropout (Srivastava et al., 2014) of rate 0.5 before the second max-pooling of each U-net, weight decay (Krogh and Hertz, 1991) of rate 10^{-5} in the stochastic gradient descent (SGD) optimizer, batch normalization (Ioffe and Szegedy, 2015) after each ReLU layer, and early stopping (Prechelt, 2002) based on the two loss functions. Algorithm 1 outlines our DeepFDR algorithm, which alternately optimizes the two loss functions. At each epoch, the algorithm updates the parameter $\boldsymbol{\theta}_1$ for $\boldsymbol{U}_1$ by minimizing the $L_{\text{soft-Ncut}}$ loss in (4), and then simultaneously updates the parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ for $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ by minimizing the L_{recon} loss in (5). After network training, the final segmentation probability map is generated using the trained network with dropout disabled, and is then passed through our LIS module to obtain the estimated LIS map $\{\widehat{LIS}_i(\boldsymbol{x})\}_{i=1}^m$ by (6). This estimated LIS map is plugged into the LIS-based testing procedure (3) to yield the multiple testing result.

3 NUMERICAL RESULTS

We compare our DeepFDR with classic and recent FDR control methods in Section 3.2 through simulations and in Section 3.3 using FDG-PET data from the Alzheimer's Disease Neuroimaging Initiative (ADNI).

3.1 Methods for Comparison

We conducted a comparative evaluation of our DeepFDR against eight existing methods, including BH (Benjamini and Hochberg, 1995), q-value (Storey et al., 2003), LocalFDR (Efron, 2004), HMRF-LIS (Shu et al., 2015), SmoothFDR (Tansey et al., 2018a), LAWS (Cai et al., 2022), NeuralFDR (Xia et al., 2017), and OrderShapeEM (OSEM; Cao et al. (2022a)). The BH, q-value, and LocalFDR methods are classic FDR control methods developed for independent tests, but HMRF-LIS, SmoothFDR, and LAWS are state-of-the-art spatial methods applicable to 3D image data. HMRF-LIS uses 1-nearest-neighbor HMRFs to model spatial dependencies and then applies the LIS-based testing procedure. SmoothFDR utilizes an empirical-Bayes approach to enforce spatial smoothness with lasso to detect localized regions of significant test statistics. LAWS constructs structure-adaptive weights based on the estimated local sparsity levels to weigh p-values. NeuralFDR is designed for multiple testing problems with covariates available, and employs a deep neural network to learn the p-value threshold as a function of the covariates; in our context, we used the 3D coordinates as three covariates for NeuralFDR. OSEM extends the LocalFDR method by incorporating auxiliary information on the order of prior null probabilities, which is often lacking in voxel-based multiple testing; to serve as the auxiliary information, q-values were employed in simulations, and both q-values and BH-adjusted p-values were attempted in the real-data analysis. The detailed implementations of the nine methods are given in Appendix.

3.2 Simulation Studies

Simulation settings. We generated each simulated dataset on a lattice cube with size $m = 30 \times 30 \times 30$. The ground-truth hypothesis state labels $\boldsymbol{h} = [h_1, \ldots, h_m]$ were generated based on the ADNI FDG-PET dataset in Section 3.3. Specifically, we used the result of the q-value method with nominal FDR level 0.01 for the comparison between the early mild cognitive impairment group and the cognitively normal group; three $30 \times 30 \times 30$ lattice cubes were randomly cropped from the brain volume of the q-value result, respectively with about 10%, 20%, and 30% of voxels tested as significant; in the three cubes, we set ground-truth values of $h_i = 1$ for the significant voxels and $h_i = 0$ for the remaining voxels. For each cube, the test statistics $\boldsymbol{x} = [x_1, \ldots, x_m]$ were generated using the Gaussian mixture model: $x_i|h_i \sim (1-h_i)N(0,1) + h_i\{\frac{1}{2}N(\mu_1, \sigma_1^2) + \frac{1}{2}N(2,1)\}$. We varied μ_1 from -4 to 0 with fixed $\sigma_1^2 = 1$, and varied σ_1^2 from 0.125 to 8 with fixed $\mu_1 = -2$. In total, we generated 45 simulation settings, including 15 different combinations of (μ_1, σ_1^2) for each of the three cubes with different proportions of signals. We conducted the nine FDR control methods with a nominal FDR level $\alpha = 0.1$ for 50 independent replications of each simulation setting. FDR, FNR, the average number of true positives (ATP), and computational time for each method were computed based on the 50 replications.

Multiple-testing results. Figures 2 and A.1-A.5 display the multiple-testing results for the three cubes with signal proportion (denoted by P_1) approximately equal to 10%, 20%, and 30%, respectively. We see that our DeepFDR well controls the FDR around the nominal level 0.1, and performs the best in 39 simulation settings and ranks second in the other 6 settings in terms of smallest FNR, largest ATP and controlled FDR. In particular, for weak signal cases where $\mu_1 \in [-2, 0]$ and $\sigma_1^2 = 1$, DeepFDR surpasses the other valid FDR control methods by a large margin. For strong signal cases with $\mu_1 \in \{-4, -3.5, -3\}$ and $\sigma_1^2 = 1$ when $P_1 \approx 10\%$ or 30%, DeepFDR is outperformed by LAWS; this behavior is reasonable since the optimality of DeepFDR's LIS-based testing procedure is asymptotic and subject to certain conditions (Sun and Cai, 2009; Xie et al., 2011). It is observed that all FDRs of NeuralFDR are more than 0.2 larger than the nominal level 0.1, OSEM and HMRF-LIS are not valid in FDR control for almost all simulation settings, and SmoothFDR is not valid for almost all settings with $P_1 \approx 10\%$ and some settings with $P_1 \approx 20\%$. This may be owing to the incompatible assumption made by NeuralFDR for spatial data (see Section 1), the failure of OSEM to consider spatial dependence, the inadequate spatial modeling by HMRF-LIS, and the oversmoothing effect of SmoothFDR. The figures show that BH, LocalFDR, and LAWS are often conservative in FDR control with FDR smaller than the nominal level with a large distance. The q-value method well controls FDR around 0.1, and has smaller FNR and larger ATP than BH and LocalFDR, but is inferior to the spatial methods LAWS and DeepFDR.

Timing performance. DeepFDR, NeuralFDR, and HMRF-LIS were executed on a NVIDIA RTX8000 GPU (48GB memory), and the other six methods were run on a server with 20 Intel Xeon Platinum 8268 CPUs (2.90GHz, 64GB memory). The computational time was computed based on the simulation setting with $(\mu_1, \sigma_1^2) = (-2, 1)$ and $P_1 \approx 20\%$. Table A.1 presents the mean and standard deviation (SD) of the runtime over the 50 simulation replications. Given that BH, q-value, and LocalFDR methods are designed for independent tests rather than spatial data, it is not surprising that they exhibit the fastest performance, each completing with a mean runtime of less than 5 seconds. Our DeepFDR boasts a mean runtime of 7.21 seconds, with an SD of 1.22 seconds, which is approximately 1.7 times the runtime of the q-value method. However, it remains notably faster than the other four

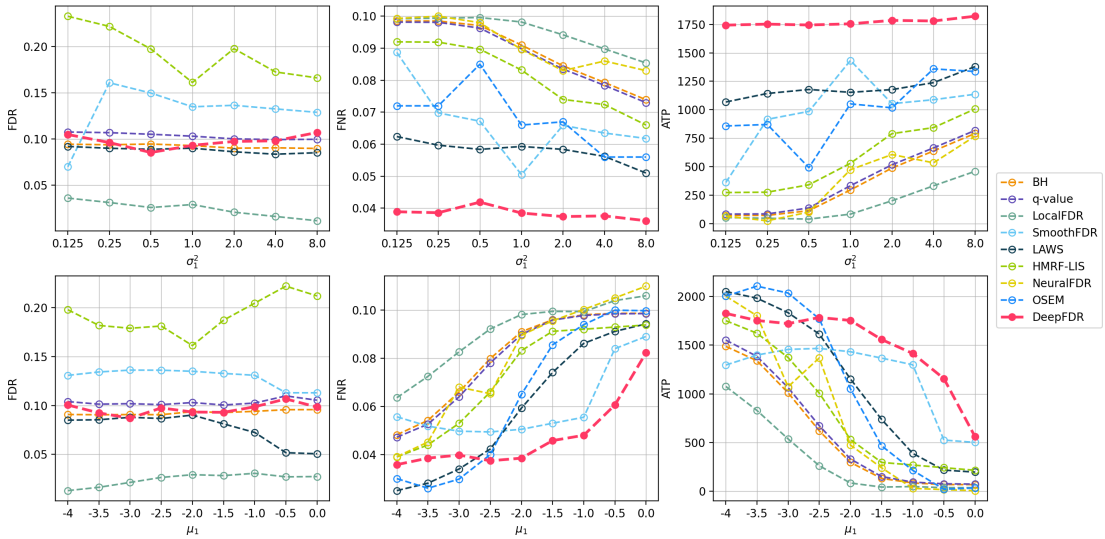

Figure 2: Simulation results for the cube with $P_1 \approx 10\%$. All FDRs of NeuralFDR and almost all FDRs of OSEM are too large, and thus their FDRs are not shown in this figure; see Figure A.3, instead.

methods, requiring only about 1/2 of the time used by OSEM, 1/8 of HMRF-LIS, 1/20 of SmoothFDR, 1/50 of LAWS, and 1/860 of NeuralFDR.

3.3 Real-data Analysis

FDG-PET is a widely used imaging technique in early diagnosis and monitoring progression of Alzheimer's disease (AD). This technique assesses brain glucose metabolism, which typically decreases in AD cases. The difference in brain glucose metabolism between two population groups can be investigated by testing the difference of their voxel-level population means in the SUVR from FDG-PET, leading to a high-dimensional spatial multiple testing problem. We employed voxel-based multiple testing methods to compare the mean SUVR difference between the cognitively normal (CN) group and each of the following three groups: early mild cognitive impairment patients with conversion to AD (EMCI2AD), late mild cognitive impairment patients with conversion to AD (LMCI2AD), and the AD group.

ADNI FDG-PET dataset. The FDG-PET image dataset used in this study was obtained from the ADNI database (adni.loni.usc.edu). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging, positron emission tomography, other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment and early AD. The dataset consists of baseline FDG-PET images from 742 subjects, including 286 CN subjects, 42 EMCI2AD patients, 175 LMCI2AD patients, and 239 AD patients. All 742 FDG-PET images were preprocessed using the Clinica software (Routier et al., 2021) to ensure spatial normalization to the MNI IXI549Space template and intensity normalization based on the average uptake value in the pons region. We considered the 120 brain regions of interest (ROIs) from the AAL2 altas (Rolls et al., 2015). The total number of voxels in the 120 ROIs is 439,758, and the number of voxels in each ROI ranges from 107 to 12,201 with a median of 2874 (see Table A.2). For each ROI voxel, we ran a linear regression with the voxel's SUVR as the response variable and the dummy variables of the EMCI2AD, LMCI2AD, and AD groups as explanatory variables (where CN was used as the reference group), adjusting for patient's age, gender, race, ethnicity, education, marital status, and APOE4 status. The voxel-level t-statistics for regression coefficients of the three groups' dummy variables and associated p-values were thus obtained for the three comparisons: EMCI2AD vs. CN, LMCI2AD vs. CN, and AD vs. CN. Z-statistics were transformed from t-statistics for certain FDR control methods that require them as input.

Multiple-testing results. All FDR control methods were conducted with the nominal FDR level $\alpha = 0.001$ for each of the three comparisons on the 439,758 ROI voxels. OSEM finds no discoveries in the three comparisons when using q-values or BH-adjusted p-values as its auxiliary information. Figures A.6–A.8 present the discoveries obtained by each method. For all methods except SmoothFDR and OSEM, it is observed that most discovered brain areas exhibit hypometabolism, and the affected areas expand and deteriorate during the AD progression from CN to EMCI2AD, then to LMCI2AD, and finally to AD. Figures A.9–A.11 show the proportion of discoveries found by each method in

each ROI for the three comparisons. The proportion of discoveries generally increases in each ROI during the AD progression, again indicating the growing impact of the disease on the brain.

In the AD vs. CN comparison, as shown in Figures A.8 and A.11, all methods, except OSEM, SmoothFDR and NeuralFDR, exhibit similar distributions for the proportion of discoveries over the 120 ROIs. SmoothFDR and NeuralFDR appear to overestimate signals, as a significant amount of their discoveries have p-values exceeding 0.001, 0.01, and 0.05 thresholds. Specifically, for SmoothFDR, NeuralFDR, and our DeepFDR, among their respective discoveries, 35.1%, 47.2%, and 2.6% have p-values > 0.001, 22.9%, 37.2%, and 0.094% have p-values > 0.01, and 12.1%, 28.5%, and 0.0096% have p-values > 0.05. For the LMCI2AD vs. CN comparison, as shown in Figures A.7 and A.10, the non-spatial methods BH, q-value, and LocalFDR are conservative in discoveries, spatial methods HMRF-LIS, LAWS, and our DeepFDR exhibit similar distributions of their discoveries, while SmoothFDR and NeuralFDR continue to demonstrate an overestimation of signals. Among the respective discoveries of SmoothFDR, NeuralFDR, and our DeepFDR, 53.6%, 66.1%, and 5.3% have p-values > 0.001, 31.2%, 52.5%, and 0.027% have p-values > 0.01, and 18.4%, 42.1%, and 0% have p-values > 0.05. This highlights the challenge of effectively controlling FDR for SmoothFDR and NeuralFDR, whereas our DeepFDR presents credible discoveries with significantly smaller p-values in the two comparisons. Note that the nominal FDR level α is 0.001, but it does not necessarily imply that a discovery with p-value slightly above 0.001 is definitively not a signal, because such thresholding of p-values does not account for the spatial dependence in neuroimaging data. However, if a discovery has a p-value much larger than the nominal level 0.001, e.g., 0.05, it is more likely to be a false discovery.

The EMCI2AD vs. CN comparison is particularly challenging among the three comparisons, yet it holds significant promise for early detection of AD. In this comparison, BH, q-value, LocalFDR, LAWS, and OSEM fail to yield any discoveries, and HMRF-LIS identifies only 3 discoveries, Indeed, there are only 101 voxels with p-values < 0.001, which reflects the difficulty of this comparison. NeuralFDR finds 14,342 discoveries, which are scattered across the brain as shown in Figures A.6 and A.9. SmoothFDR identifies 86,719 discoveries, but the result appears oversmoothed as shown in Figure A.6. NeuralFDR and SmoothFDR seem to overestimate the signals, with 95.7% and 68.1% of their discoveries having p-values > 0.05. In contrast, DeepFDR provides 1087 discoveries, of which 82 are among the 101 voxels with p-values < 0.001. Impressively, 88.9%, 99.3%, and 100% of DeepFDR's discoveries have p-values less than 0.005, 0.01, and 0.05, respectively. All of DeepFDR's discoveries are located in the left hemisphere, with 1080 of them found in left parahippocampal gyrus (n=276, P=11.85%), left hippocampus (n=130, P=5.84%), left inferior temporal gyrus (n=392, P=5.18%), left middle temporal gyrus (n=244, P=2.08%), and left fusiform gyrus (n=38, P=0.70%). This aligns with prior research suggesting greater vulnerability of the left hemisphere to AD (Thompson et al., 2001, 2003; Roe et al., 2021). These five ROIs are known to be early affected by AD (Echávarri et al., 2011; Braak et al., 1993; Convit et al., 2000), providing additional support for the validity of DeepFDR's discoveries.

Timing performance. We executed the methods using the same computational resource as specified in Section 3.2. Table A.1 shows the mean and SD of the runtime over the three comparisons for the ADNI FDG-PET data. The three non-spatial methods BH, q-value and LocalFDR exhibit dominant performance. Our DeepFDR follows closely in efficiency; it averaged a runtime of of 89.98 seconds with an SD of 5.17 seconds, which is merely 1.31 times the runtime of the q-value method. In stark contrast, the mean runtime for each of the other five methods exceeds 5 hours, with LAWS taking nearly 7 days. These results emphasize the high computational efficiency of our DeepFDR when tackling the voxel-based multiple testing challenge in neuroimaging data analysis.

4 CONCLUSION

This paper proposes DeepFDR, a novel deep learning-based FDR control method for voxel-based multiple testing. DeepFDR harnesses deep learning-based unsupervised image segmentation, specifically a modified W-net, to effectively capture spatial dependencies among voxel-based tests, and then utilizes the LIS-based testing procedure to achieve FDR control and minimize the FNR. Our extensive numerical studies, including comprehensive simulations and in-depth analysis of 3D FDG-PET images related to Alzheimer's disease, corroborate DeepFDR's superiority over existing methods. DeepFDR consistently demonstrates its ability to effectively control the FDR while substantially reducing the FNR, thereby enhancing the overall reliability of results in neuroimaging studies. Furthermore, DeepFDR distinguishes itself by its remarkable computational efficiency. By leveraging well-established software and advanced optimization algorithms from the field of deep learning, it stands as an exceptionally fast and efficient solution for addressing the voxel-based multiple testing problem in large-scale neuroimaging data analysis.

Acknowledgements

Dr. Shu's research was partially supported by the grant R21AG070303 from the National Institutes of Health (NIH). Dr. de Leon's research was partially supported by the NIH grants AG022374, AG12101, AG13616, AG057570, AG057848, and AG058913. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

* Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

References

Alexander, G. E., Chen, K., Pietrini, P., Rapoport, S. I., and Reiman, E. M. (2002). Longitudinal pet evaluation of cerebral metabolic decline in dementia: a potential outcome measure in alzheimer's disease treatment studies. *American Journal of Psychiatry*, 159(5):738–745.

Ashburner, J. and Friston, K. J. (2000). Voxel-based morphometry—the methods. *Neuroimage*, 11(6):805–821.

Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R. T., Berger, C., Ha, S. M., Rozycki, M., et al. (2019). Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint arXiv:1811.02629v3*.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300.

Braak, H., Braak, E., and Bohl, J. (1993). Staging of alzheimer-related cortical destruction. *European neurology*, 33(6):403–408.

Brodmann, K. (2007). *Brodmann's: Localisation in the cerebral cortex*. Springer Science & Business Media.

Buhmann, J. M. and Held, M. (1999). Unsupervised learning without overfitting: Empirical risk approximation as an induction principle for reliable clustering. In *International Conference on Advances in Pattern Recognition: Proceedings of ICAPR'98, 23–25 November 1998, Plymouth, UK*, pages 167–176. Springer.

Cai, T. T., Sun, W., and Xia, Y. (2022). Laws: A locally adaptive weighting and screening approach to spatial multiple testing. *Journal of the American Statistical Association*, 117(539):1370–1383.

Cao, H., Chen, J., and Zhang, X. (2022a). Optimal false discovery rate control for large scale multiple testing with auxiliary information. 50:807–857.

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., and Wang, M. (2022b). Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer.

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A. L., and Zhou, Y. (2021). Transunet: Transformers make strong encoders

for medical image segmentation. *arXiv preprint arXiv:2102.04306*.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. (2016). 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer.

Convit, A., De Asis, J., De Leon, M., Tarshish, C., De Santi, S., and Rusinek, H. (2000). Atrophy of the medial occipitotemporal, inferior, and middle temporal gyri in non-demented elderly predict decline to alzheimer's disease. *Neurobiology of aging*, 21(1):19–26.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Drzezga, A., Lautenschlager, N., Siebner, H., Riemenschneider, M., Willoch, F., Minoshima, S., Schwaiger, M., and Kurz, A. (2003). Cerebral metabolic changes accompanying conversion of mild cognitive impairment into alzheimer's disease: a pet follow-up study. *European Journal of Nuclear Medicine and Molecular Imaging*, 30(8):1104–1113.

Echávarri, C., Aalten, P., Uylings, H. B., Jacobs, H., Visser, P. J., Gronenschild, E., Verhey, F., and Burgmans, S. (2011). Atrophy in the parahippocampal gyrus as an early biomarker of alzheimer's disease. *Brain Structure and Function*, 215:265–271.

Efron, B. (2004). Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *Journal of the American Statistical Association*, 99(465):96–104.

Genovese, C. and Wasserman, L. (2002). Operating characteristics and extensions of the false discovery rate procedure. *Journal of the Royal Statistical Society: Series B*, 64(3):499–517.

Genovese, C. R., Lazar, N. A., and Nichols, T. (2002). Thresholding of statistical maps in functional neuroimaging using the false discovery rate. *Neuroimage*, 15(4):870–878.

Han, Y., Wang, Y., and Wang, Z. (2023). A spatially adaptive large-scale multiple testing procedure. *Stat*, page e565.

Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H. R., and Xu, D. (2022). Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 574–584.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr.

Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., and Maier-Hein, K. H. (2021). nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211.

Kanezaki, A. (2018). Unsupervised image segmentation by backpropagation. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1543–1547. IEEE.

Kantarci, K., Boeve, B. F., Przybelski, S. A., Lesnick, T. G., Chen, Q., Fields, J., Schwarz, C. G., Senjem, M. L., Gunte, J. L., Jack, C. R., et al. (2021). Fdg pet metabolic signatures distinguishing prodromal dlb and prodromal ad. *NeuroImage: Clinical*, 31:102754.

Kim, J., Yu, D., Lim, J., and Won, J.-H. (2018). A peeling algorithm for multiple testing on a random field. *Computational Statistics*, 33:503–525.

Kim, W., Kanezaki, A., and Tanaka, M. (2020). Unsupervised learning of image segmentation based on differentiable feature clustering. *IEEE Transactions on Image Processing*, 29:8055–8068.

Krogh, A. and Hertz, J. (1991). A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4:950–957.

Lee, D., Kang, H., Kim, E., Lee, H., Kim, H., Kim, Y. K., Lee, Y., and Lee, D. S. (2015). Optimal likelihood-ratio multiple testing with application to alzheimer's disease and questionable dementia. *BMC Medical Research Methodology*, 15(1):9.

Liew, S.-L., Lo, B. P., Donnelly, M. R., Zavaliangos-Petropulu, A., Jeong, J. N., Barisano, G., Hutton, A., Simon, J. P., Juliano, J. M., Suri, A., et al. (2022). A large, curated, open-source stroke neuroimaging dataset to improve lesion segmentation algorithms. *Scientific data*, 9(1):320.

Liu, J., Zhang, C., and Page, D. (2016). Multiple testing under dependence via graphical models. *The Annals of Applied Statistics*, 10(3):1699 – 1724.

Liu, Y., Yu, C., Zhang, X., Liu, J., Duan, Y., Alexander-Bloch, A. F., Liu, B., Jiang, T., and Bullmore, E. (2013). Impaired long distance functional connectivity and weighted network architecture in alzheimer's disease. *Cerebral Cortex*, 24(6):1422–1435.

Marandon, A., Lei, L., Mary, D., and Roquain, E. (2022). Machine learning meets false discovery rate. *arXiv preprint arXiv:2208.06685*.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542.

Mirman, D., Landrigan, J.-F., Kokolis, S., Verillo, S., Ferrara, C., and Pustina, D. (2018). Corrections for multiple comparisons in voxel-based lesion-symptom mapping. *Neuropsychologia*, 115:112–123.

Mosconi, L., Tsui, W.-H., De Santi, S., Li, J., Rusinek, H., Convit, A., Li, Y., Boppana, M., and De Leon, M. (2005). Reduced hippocampal metabolism in mci and ad: automated fdg-pet image analysis. *Neurology*, 64(11):1860–1867.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Ou, Y.-N., Xu, W., Li, J.-Q., Guo, Y., Cui, M., Chen, K.-L., Huang, Y.-Y., Dong, Q., Tan, L., and Yu, J.-T. (2019). Fdg-pet as an independent biomarker for alzheimer's biological diagnosis: a longitudinal study. *Alzheimer's Research & Therapy*, 11(1):57.

Pan, S., Liu, X., Xie, N., and Chong, Y. (2023). Eg-transunet: a transformer-based u-net with enhanced and guided models for biomedical image segmentation. *BMC bioinformatics*, 24(1):85.

Prechelt, L. (2002). Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer.

Pu, Y., Sun, J., Tang, N., and Xu, Z. (2023). Deep expectation-maximization network for unsupervised image segmentation and clustering. *Image and Vision Computing*, 135:104717.

Roe, J. M., Vidal-Piñeiro, D., Sørensen, Ø., Brandmaier, A. M., Düzel, S., Gonzalez, H. A., Kievit, R. A., Knights, E., Kühn, S., Lindenberger, U., et al. (2021). Asymmetric thinning of the cerebral cortex across the adult lifespan is accelerated in alzheimer's disease. *Nature communications*, 12(1):721.

Rolls, E. T., Joliot, M., and Tzourio-Mazoyer, N. (2015). Implementation of a new parcellation of the orbitofrontal cortex in the automated anatomical labeling atlas. *Neuroimage*, 122:1–5.

Romano, Y., Sesia, M., and Candès, E. (2020). Deep knockoffs. *Journal of the American Statistical Association*, 115(532):1861–1872.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Routier, A., Burgos, N., Díaz, M., Bacci, M., Bottani, S., El-Rifai, O., Fontanella, S., Gori, P., Guillon, J., Guyot, A., et al. (2021). Clinica: An open-source software platform for reproducible clinical neuroscience studies. *Frontiers in Neuroinformatics*, 15:689675.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.

Shivamurthy, V. K., Tahari, A. K., Marcus, C., and Subramaniam, R. M. (2015). Brain fdg pet and the diagnosis of dementia. *American Journal of Roentgenology*, 204(1):W76–W85.

Shu, H., Nan, B., and Koeppe, R. (2015). Multiple testing for neuroimaging via hidden markov random field. *Biometrics*, 71(3):741–750.

Siddique, N., Paheding, S., Elkin, C. P., and Devabhaktuni, V. (2021). U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access*, 9:82031–82057.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Storey, J. D. et al. (2003). The positive false discovery rate: a bayesian interpretation and the q-value. *The Annals of Statistics*, 31(6):2013–2035.

Sun, W. and Cai, T. T. (2009). Large-scale multiple testing under dependence. *Journal of the Royal Statistical Society: Series B*, 71(2):393–424.

Sun, W., Reich, B. J., Cai, T. T., Guindani, M., and Schwartzman, A. (2015). False discovery control in large-scale spatial multiple testing. *Journal of the Royal Statistical Society: Series B*, 77(1):59–83.

Sun, Y., Gao, K., Wu, Z., Li, G., Zong, X., Lei, Z., Wei, Y., Ma, J., Yang, X., Feng, X., et al. (2021). Multi-site infant brain segmentation algorithms: the iseg-2019 challenge. *IEEE Transactions on Medical Imaging*, 40(5):1363–1376.

Tansey, W., Koyejo, O., Poldrack, R. A., and Scott, J. G. (2018a). False discovery rate smoothing. *Journal of the American Statistical Association*, 113(523):1156–1171.

Tansey, W., Wang, Y., Blei, D., and Rabadan, R. (2018b). Black box fdr. In *International conference on machine learning*, pages 4867–4876. PMLR.

Thompson, P. M., Hayashi, K. M., De Zubicaray, G., Janke, A. L., Rose, S. E., Semple, J., Herman, D., Hong, M. S., Dittmer, S. S., Doddrell, D. M., et al. (2003). Dynamics of gray matter loss in alzheimer's disease. *Journal of neuroscience*, 23(3):994–1005.

Thompson, P. M., Mega, M. S., Woods, R. P., Zoumalan, C. I., Lindshield, C. J., Blanton, R. E., Moussai, J., Holmes, C. J., Cummings, J. L., and Toga, A. W. (2001). Cortical change in alzheimer's disease detected with a disease-specific population-based brain atlas. *Cerebral Cortex*, 11(1):1–16.

Xia, F., Zhang, M. J., Zou, J. Y., and Tse, D. (2017). Neuralfdr: Learning discovery thresholds from hypothesis features. *Advances in neural information processing systems*, 30.

Xia, X. and Kulis, B. (2017). W-net: A deep model for fully unsupervised image segmentation. *arXiv preprint arXiv:1711.08506*.

Xie, J., Cai, T. T., Maris, J., and Li, H. (2011). Optimal false discovery rate control for dependent data. *Statistics and its interface*, 4(4):417–430.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [No. The code is available at https://github.com/kimtae55/DeepFDR. The FDG-PET image dataset used in our paper was obtained from the Alzheimer's Disease Neuroimaging Initiative (https://adni.loni.usc.edu). We are not allowed to share this dataset. The instructions have been given in the paper and its appendix.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes. If applicable, the license information for existing codes can be found at the URLs provided in our paper.]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Yes. We have successfully executed the data use agreement with our data provider, the Alzheimer's Disease Neuroimaging Initiative (ADNI). We have included an acknowledgment to ADNI in this paper.]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

APPENDIX

A.1 Implementation Details of Comparison Methods

In this section, we provide a comprehensive overview of the implementation details for all the methods used in our numerical comparisons. It is worth noting that the Python versions of the methods consistently demonstrated superior speed compared to their R counterparts. Thus, we prioritized Python versions whenever available, only resorting to R when necessary. Our numerical studies, including simulations and real-data analysis, were conducted using Python 3.9.7 and R 4.2.1.

BH and LocalFDR: We used the Python package **statsmodels** (v0.12.2) available at `https://www.statsmodels.org`.

q-value: We used the Python package **multipy** (v0.16) available at `https://github.com/puolival/multipy`.

HMRF-LIS: The original implementation is in C++ (`https://github.com/shu-hai/FDRhmrf`), but its sequential nature using Gibbs sampling poses scalability challenges. To address this, we have created a Python version that utilizes GPU-based HMRF Gibbs sampling. Although Gibbs sampling is traditionally sequential, the Ising model-based HMRF used by the method exhibits a dependency on neighboring voxels that can be parallelized by modeling the input voxels as a black and white checkerboard. We applied a convolutional operation with a suitable $3 \times 3 \times 3$ kernel to extract information from neighboring voxels, achieving significant speedup and faster convergence. In simulations, we used a single HMRF to model the $30 \times 30 \times 30$ lattice cube. But in real-data analysis, we modeled each ROI with a separate HMRF, following the HMRF-LIS paper.

SmoothFDR: We utilized the author-published Python package available at `https://github.com/tansey/smoothfdr`, using 20 sweeps.

NeuralFDR: We utilized the author-published Python package available at `https://github.com/fxia22/NeuralFDR/tree/master`. The input consists of each test's p-value and corresponding covariates. We used the 3D coordinates as the covariates. We noticed the standard practice of mini-batch based training was not implemented in their code, resulting in GPU memory allocation issues when handling the ADNI data. Thus, we modified their code to incorporate mini-batches during the forward pass, aggregating the respective losses instead of inputting the entire training set at once. For simulations, we used the default parameters in their code, but in real-data analysis, we set n-init=3 and num-iterations=200 to reduce the computational time.

LAWS: Only R code is available in the Supplementary Materials of its paper at `https://doi.org/10.1080/01621459.2020.1859379`. The 3D implementation of LAWS was used in both simulations and real-data analysis.

OrderShapeEM (OSEM): We used the author-published R package available at `https://github.com/jchen1981/OrderShapeEM`. To serve as the auxiliary information on the order of prior null probabilities, q-values were employed in simulations, and both q-values and BH-adjusted p-values were attempted in the real-data analysis.

DeepFDR: We implemented our algorithm using the Pytorch package (v2.0.1) for the network. The code is available at `https://github.com/kimtae55/DeepFDR`. Most details can be found in Section 2.3 of our paper. For training, the SGD optimizer with a momentum of 0.9 and weight decay of 10^{-5} was used with Kaiming initialization for weights. The learning rate was tuned and early stopping was applied based on the two loss functions. The best learning rate was 0.05 for most simulation settings and 0.07 for the others, and is 0.008, 0.001, and 0.006 for EMCI2AD vs CN, LMCI2AD vs CN, and AD vs CN, respectively. The algorithm was terminated before 25 epochs for all simulation settings and 10 epochs for all comparisons in real-data analysis. In a preliminary simulation, the parameters $(\sigma_x, \sigma_\ell, r)$ were slightly tuned around the values (10,4,5) used by Xia and Kulis (2017). Despite this fine-tuning not significantly altering the results, these parameters were ultimately set to (11,3,3) for the final simulations and real-data analysis.

BH and q-value methods take a 1D sequence of p-values as input, OSEM requires a 1D sequence of p-values and a 1D sequence of auxiliary information on the order of prior null probabilities, NeuralFDR accepts p-values and 3D coordinates as input, LAWS takes a 3D volume of p-values, LocalFDR requires a 1D sequence of z-values, HMFR-LIS and SmoothFDR expect a 3D volume of z-values, and DeepFDR takes a 3D volume of test statistics (z-values in simulations and t-values in real-data analysis) and the corresponding 3D volume of p-values as input. In simulations, the 3D volume had a size of $30 \times 30 \times 30$ given to the other methods, and DeepFDR zero-padded

the volume to size $32 \times 32 \times 32$ to facilitate the two max-pooling layers in each U-net of its network. In real-data analysis, the 3D volume was cropped to size $100 \times 120 \times 100$ from the original brain image size of $121 \times 145 \times 121$ by removing redundant background voxels; the non-ROI voxels were set with 0 for t-values and z-values, and 1 for p-values; only tests on the ROI voxels were used to yield the multiple testing results.

A.2 Supplementary Tables and Figures for Numerical Results

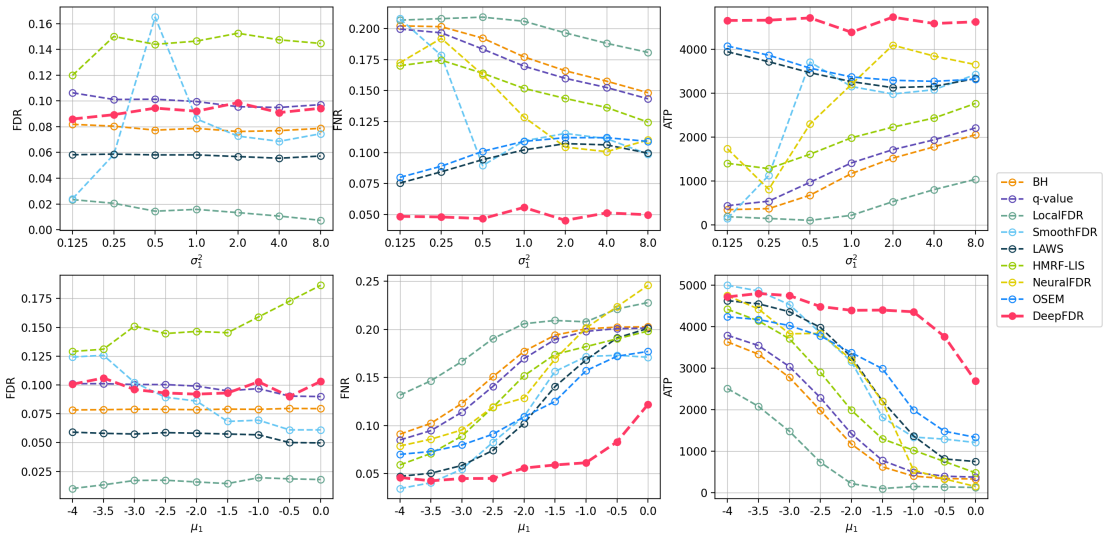

Figure A.1: Simulation results for the cube with $P_1 \approx 20\%$. FDRs for NeuralFDR and OSEM are too large and are thus not shown in this figure; see Figure A.4, instead.

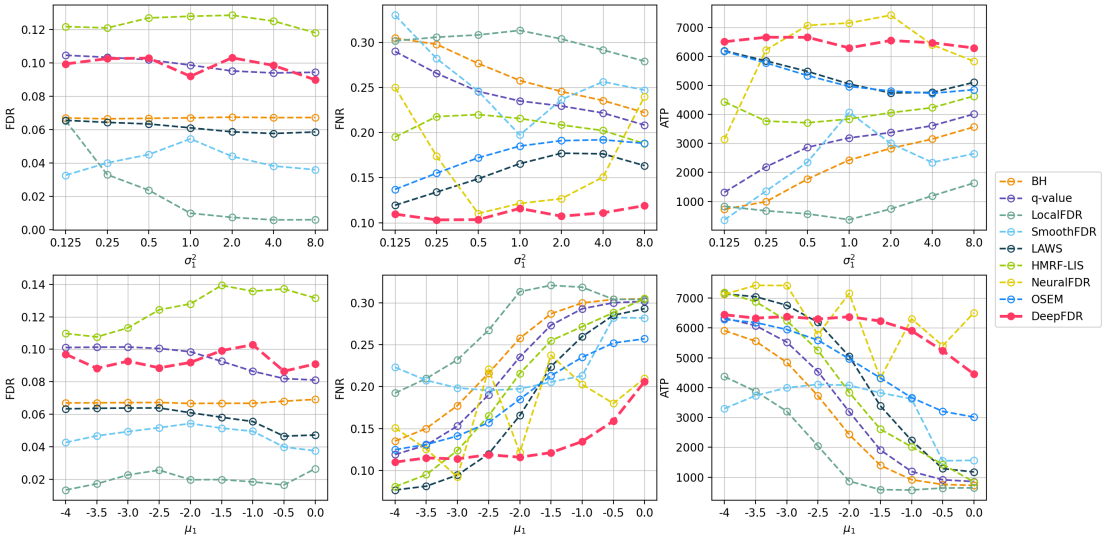

Figure A.2: Simulation results for the cube with $P_1 \approx 30\%$. FDRs for NeuralFDR and OSEM are too large and are thus not shown in this figure; see Figure A.5, instead.

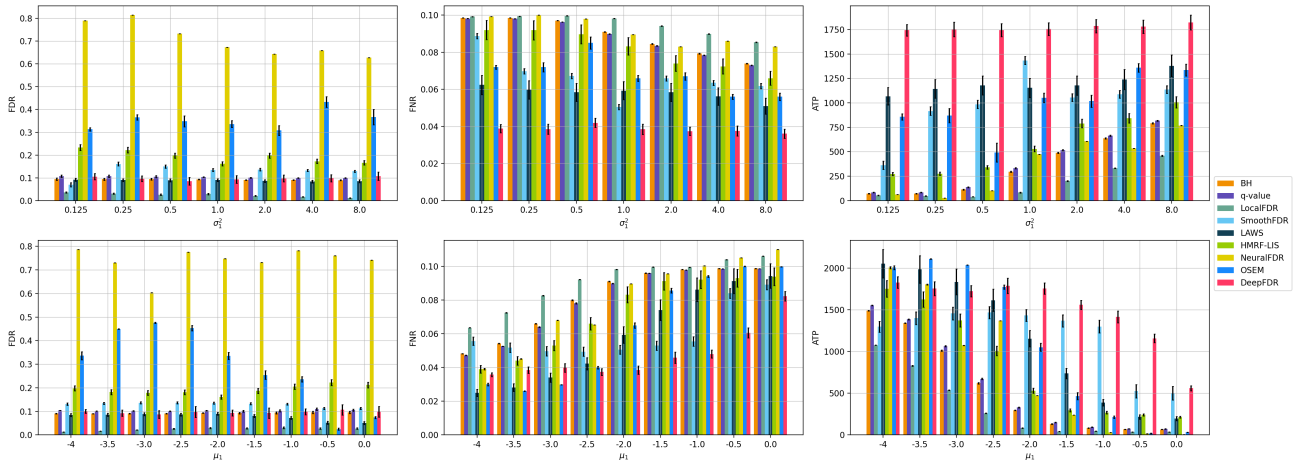

Figure A.3: Simulation results with standard error bars for the cube with $P_1 \approx 10\%$.

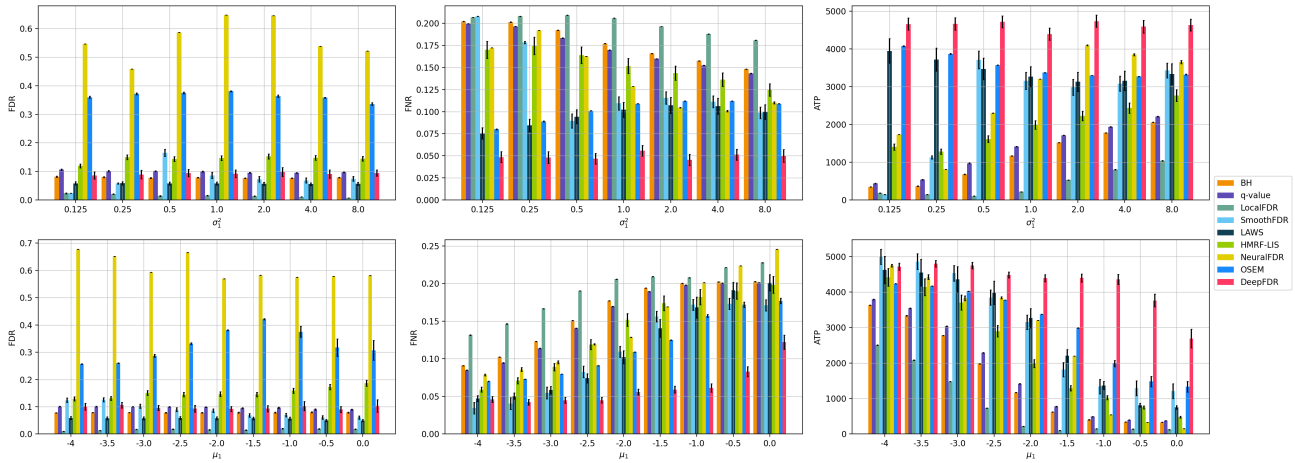

Figure A.4: Simulation results with standard error bars for the cube with $P_1 \approx 20\%$.

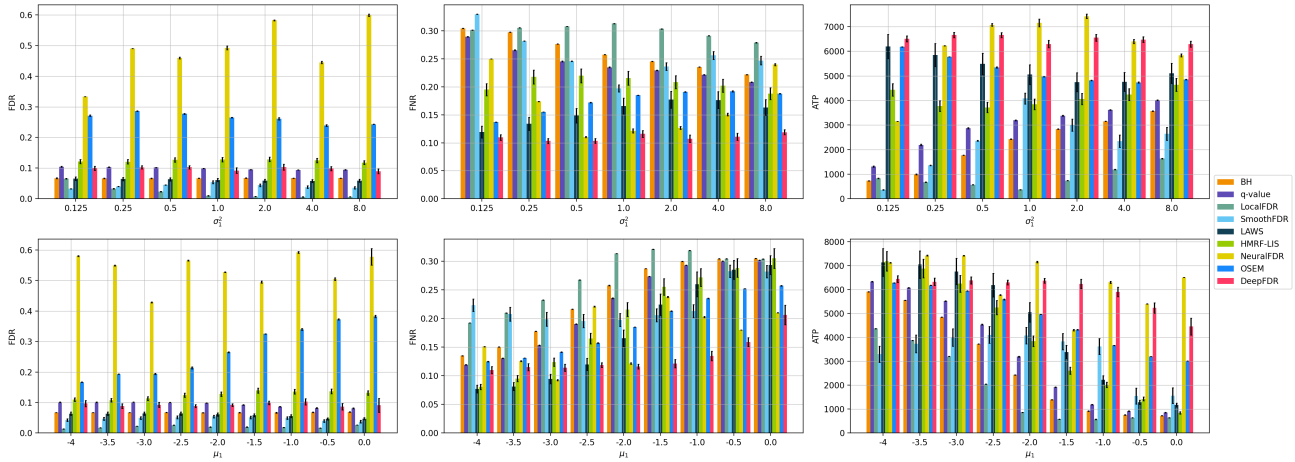

Figure A.5: Simulation results with standard error bars for the cube with $P_1 \approx 30\%$.

Method	Simulation	ADNI data
BH	0.0794 (0.0088)	0.1320 (0.0266)
q-value	4.2547 (0.0422)	68.458 (0.3169)
LocalFDR	0.1969 (0.0119)	0.4005 (0.0726)
SmoothFDR	143.92 (2.0182)	53281 (3437.2)
LAWS	371.49 (1.2788)	611620 (24745)
HMRF-LIS	56.932 (6.2486)	20245 (1987.0)
NeuralFDR	6205.1 (412.93)	95388 (6198.1)
OSEM	15.565 (5.2412)	93312 (21603)
DeepFDR	7.2104 (1.2248)	89.984 (5.1672)

Table A.1: Mean (and SD) of runtime in seconds.

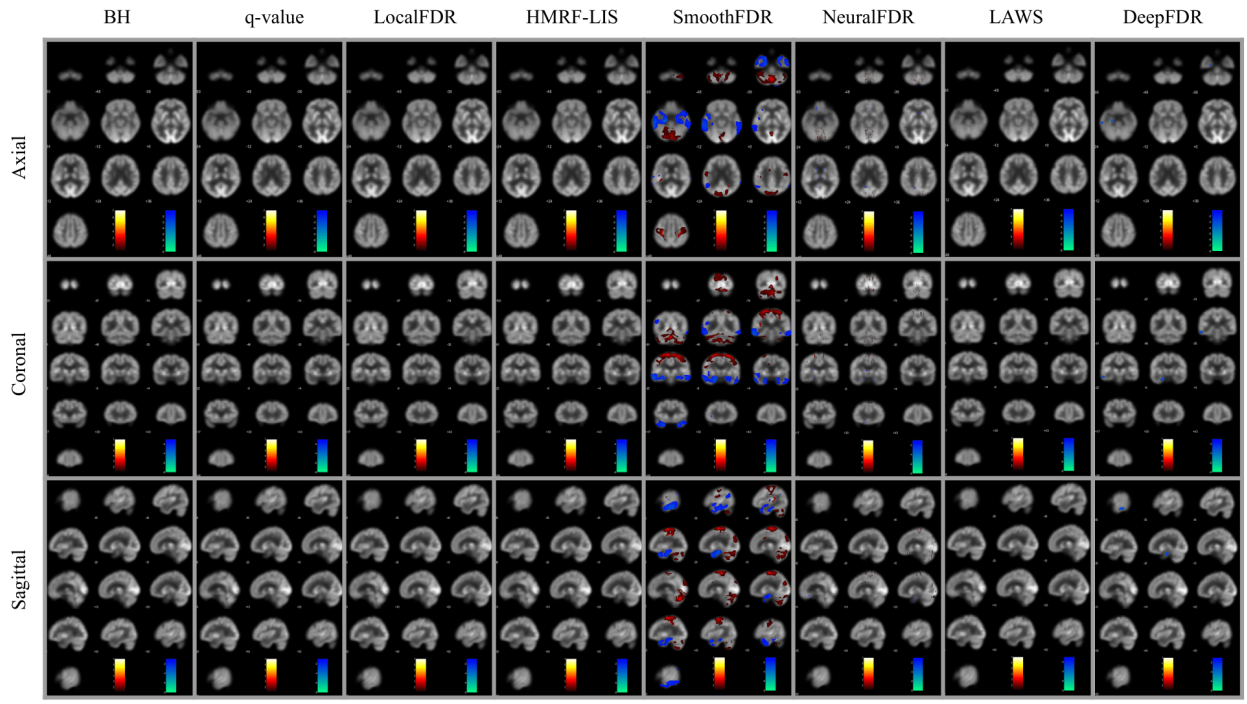

Figure A.6: Z-statistics of the discoveries by each considered method for EMCI2AD vs. CN. OSEM found no discoveries and is thus omitted.

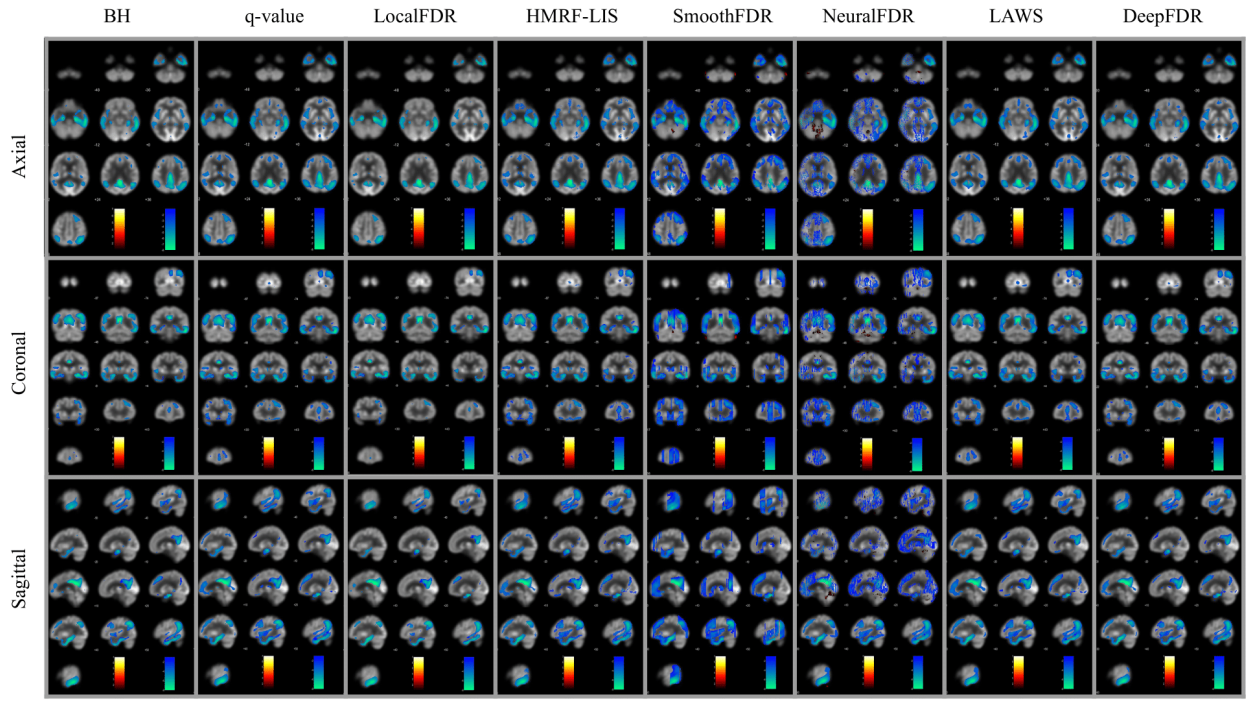

Figure A.7: Z-statistics of the discoveries by each considered method for LMCI2AD vs. CN. OSEM found no discoveries and is thus omitted.

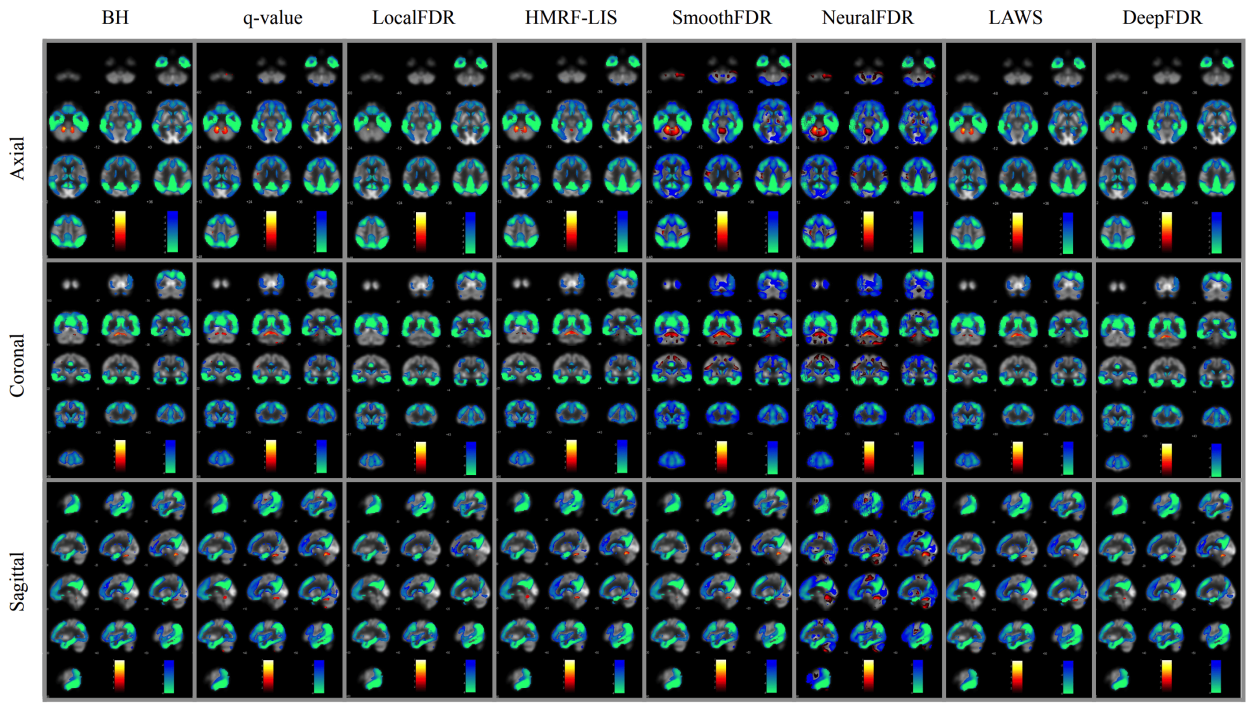

Figure A.8: Z-statistics of the discoveries by each considered method for AD vs. CN. OSEM found no discoveries and is thus omitted.

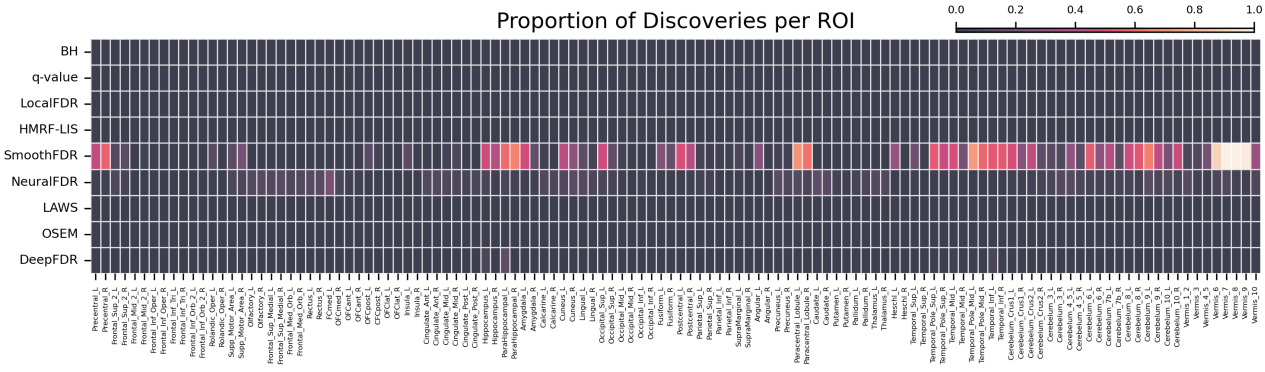

Figure A.9: Heatmap illustrating the proportion of discoveries in each ROI for EMCI2AD vs. CN.

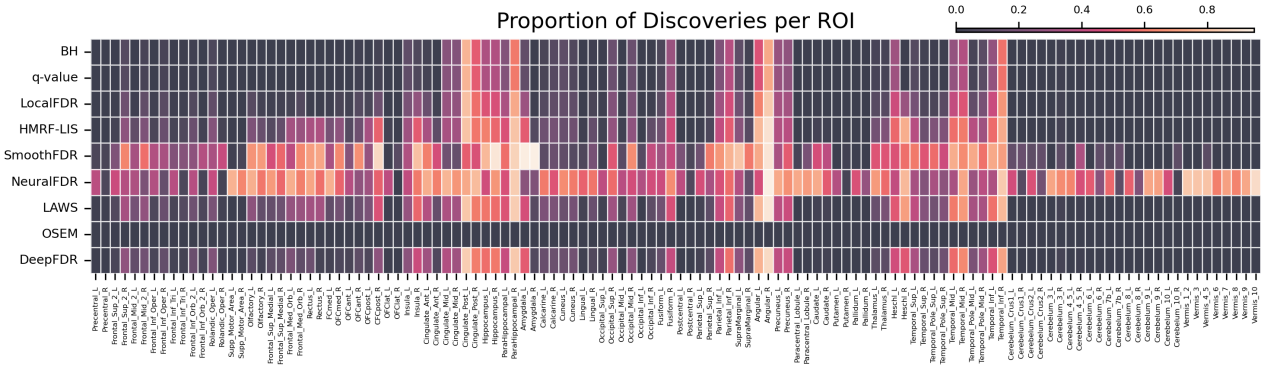

Figure A.10: Heatmap illustrating the proportion of discoveries in each ROI for LMCI2AD vs. CN.

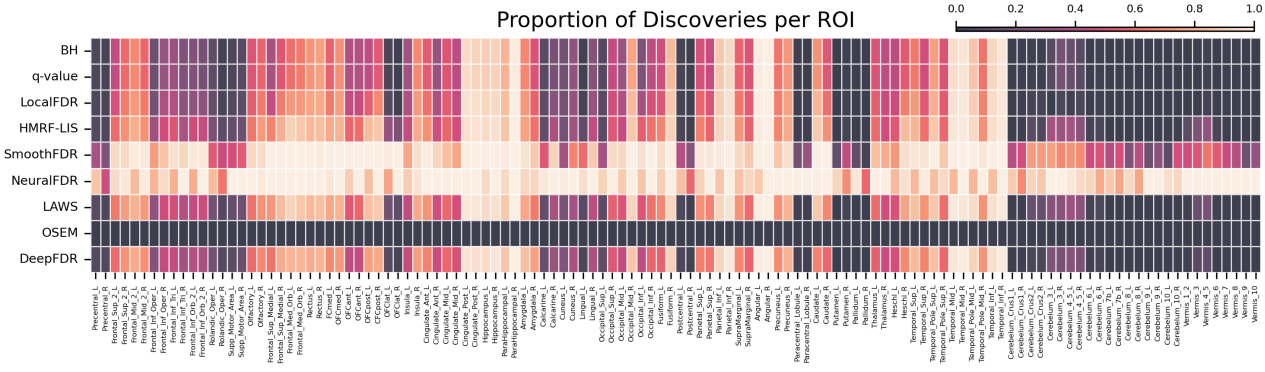

Figure A.11: Heatmap illustrating the proportion of discoveries in each ROI for AD vs. CN.

ROI	# voxels	ROI	# voxels	ROI	# voxels
Precentral_L	8281	Precentral_R	7972	Frontal_Sup_2_L	11315
Frontal_Sup_2_R	12201	Frontal_Mid_2_L	10701	Frontal_Mid_2_R	11617
Frontal_Inf_Oper_L	2496	Frontal_Inf_Oper_R	3303	Frontal_Inf_Tri_L	6020
Frontal_Inf_Tri_R	5213	Frontal_Inf_Orb_2_L	1754	Frontal_Inf_Orb_2_R	1877
Rolandic_Oper_L	2405	Rolandic_Oper_R	3210	Supp_Motor_Area_L	5057
Supp_Motor_Area_R	5861	Olfactory_L	648	Olfactory_R	726
Frontal_Sup_Medial_L	7178	Frontal_Sup_Medial_R	4881	Frontal_Med_Orb_L	1793
Frontal_Med_Orb_R	2176	Rectus_L	1950	Rectus_R	1759
FCmed_L	1272	OFCmed_R	1457	OFCant_L	1137
OFCant_R	1631	OFCpost_L	1410	CFCpost_R	1401
OFClat_L	488	OFClat_R	475	Insula_L	4418
Insula_R	4204	Cingulate_Ant_L	3289	Cingulate_Ant_R	3230
Cingulate_Mid_L	4487	Cingulate_Mid_R	5169	Cingulate_Post_L	1079
Cingulate_Post_R	767	Hippocampus_L	2225	Hippocampus_R	2265
ParaHippocampal_L	2330	ParaHippocampal_R	2675	Amygdala_L	504
Amygdala_R	599	Calcarine_L	5392	Calcarine_R	4473
Cuneus_L	3716	Cuneus_R	3291	Lingual_L	4945
Lingual_R	5398	Occipital_Sup_L	3179	Occipital_Sup_R	3382
Occipital_Mid_L	7876	Occipital_Mid_R	4865	Occipital_Inf_L	2133
Occipital_Inf_R	2401	Fusiform_L	5410	Fusiform_R	5976
Postcentral_L	9295	Postcentral_R	9045	Parietal_Sup_L	4853
Parietal_Sup_R	5234	Parietal_Inf_L	5753	Parietal_Inf_R	3221
SupraMarginal_L	2961	SupraMarginal_R	4536	Angular_L	2786
Angular_R	4129	Precuneus_L	8253	Precuneus_R	7862
Paracentral_Lobule_L	3217	Paracentral_Lobule_R	2035	Caudate_L	2280
Caudate_R	2377	Putamen_L	2392	Putamen_R	2532
Pallidum_L	665	Pallidum_R	635	Thalamus_L	2667
Thalamus_R	2600	Heschl_L	525	Heschl_R	579
Temporal_Sup_L	5641	Temporal_Sup_R	7547	Temporal_Pole_Sup_L	3005
Temporal_Pole_Sup_R	3162	Temporal_Mid_L	11745	Temporal_Mid_R	10556
Temporal_Pole_Mid_L	1789	Temporal_Pole_Mid_R	2786	Temporal_Inf_L	7562
Temporal_Inf_R	8339	Cerebelum_Crus1_L	6152	Cerebelum_Crus1_R	6258
Cerebelum_Crus2_L	4522	Cerebelum_Crus2_R	4994	Cerebelum_3_L	334
Cerebelum_3_R	536	Cerebelum_4_5_L	2747	Cerebelum_4_5_R	2086
Cerebelum_6_L	4113	Cerebelum_6_R	4291	Cerebelum_7b_L	1388
Cerebelum_7b_R	1276	Cerebelum_8_L	4454	Cerebelum_8_R	5490
Cerebelum_9_L	2069	Cerebelum_9_R	1956	Cerebelum_10_L	328
Cerebelum_10_R	374	Vermis_1_2	107	Vermis_3	492
Vermis_4_5	1442	Vermis_6	766	Vermis_7	468
Vermis_8	512	Vermis_9	412	Vermis_10	284

Table A.2: The number of voxels in each ROI.

Method	PHL	HL	TIL	TML	FL	TPML	TPSL	PREL	PRER	FS2L
BH	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
q-value	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
LocalFDR	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
HMRF-LIS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000
SmoothFDR	0.6009	0.4710	0.5440	0.4928	0.2396	0.7289	0.5524	0.4268	0.5696	0.0860
NeuralFDR	0.0476	0.0521	0.0057	0.0066	0.0043	0.0028	0.0027	0.0085	0.0103	0.0675
LAWS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
OSEM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DeepFDR	0.1185	0.0584	0.0518	0.0208	0.0070	0.0028	0.0007	0.0000	0.0000	0.0000

Table A.3: Proportion of discoveries in the top 10 affected ROIs detected by DeepFDR for EMCI2AD vs. CN. See Table A.6 for region abbreviations.

Method	ANR	CPL	PHR	ANL	TIR	MTR	CPR	HR	PIR	HL
BH	0.7266	0.7618	0.6064	0.4856	0.5864	0.3633	0.5254	0.3620	0.3772	0.3537
q-value	0.7266	0.7618	0.6064	0.4856	0.5864	0.3633	0.5254	0.3620	0.3772	0.3537
LocalFDR	0.8302	0.7998	0.7338	0.6242	0.6775	0.4941	0.5763	0.4773	0.4617	0.4921
HMRF-LIS	0.9026	0.8054	0.8090	0.7757	0.7349	0.6106	0.5997	0.5545	0.5253	0.6225
SmoothFDR	0.9121	0.5329	0.8561	0.7297	0.6998	0.6509	0.4811	0.9161	0.7566	0.7766
NeuralFDR	0.9489	0.7294	0.8348	0.4648	0.8274	0.7230	0.7836	0.6777	0.5709	0.4512
LAWS	0.9157	0.8174	0.8378	0.7721	0.7754	0.6852	0.6115	0.5744	0.5473	0.5960
OSEM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DeepFDR	0.8762	0.8378	0.8262	0.7538	0.7152	0.6438	0.6141	0.6031	0.5815	0.5748

Table A.4: Proportion of discoveries in the top 10 affected ROIs detected by DeepFDR for LMCI2AD vs. CN. See Table A.6 for region abbreviations.

Method	ANR	PHR	AL	TMR	TIR	PIR	TML	HR	CPL	TIL
BH	1.0000	0.9869	0.9871	0.9605	0.9621	0.9255	0.9367	0.8971	0.9323	0.9162
q-value	1.0000	0.9869	0.9871	0.9605	0.9621	0.9255	0.9367	0.8971	0.9323	0.9162
LocalFDR	1.0000	0.9918	0.9896	0.9737	0.9704	0.9419	0.9537	0.9227	0.9527	0.9312
HMRF-LIS	1.0000	0.9940	0.9878	0.9798	0.9734	0.9497	0.9658	0.9426	0.9425	0.9378
SmoothFDR	1.0000	1.0000	1.0000	1.0000	0.9999	1.0000	0.9999	1.0000	1.0000	0.9985
NeuralFDR	1.0000	1.0000	0.8726	1.0000	1.0000	1.0000	0.8163	1.0000	1.0000	0.8360
LAWS	1.0000	0.9963	0.9910	0.9673	0.9797	0.9581	0.9743	0.9435	0.9731	0.9501
OSEM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DeepFDR	1.0000	0.9981	0.9910	0.9821	0.9800	0.9733	0.9726	0.9660	0.9592	0.9528

Table A.5: Proportion of discoveries in the top 10 affected ROIs detected by DeepFDR for AD vs. CN. See Table A.6 for region abbreviations.

Code	ROI Name	Code	ROI Name	Code	ROI Name
PREL	Precentral_L	PRER	Precentral_R	FS2L	Frontal_Sup_2_L
FS2R	Frontal_Sup_2_R	FM2L	Frontal_Mid_2_L	FM2R	Frontal_Mid_2_R
FIOL	Frontal_Inf_Oper_L	FIOR	Frontal_Inf_Oper_R	FITL	Frontal_Inf_Tri_L
FITR	Frontal_Inf_Tri_R	FIO2L	Frontal_Inf_Orb_2_L	FIO2R	Frontal_Inf_Orb_2_R
ROL	Rolandic_Oper_L	ROR	Rolandic_Oper_R	SMAL	Supp_Motor_Area_L
SMAR	Supp_Motor_Area_R	OL	Olfactory_L	OR	Olfactory_R
FSML	Frontal_Sup_Medial_L	FSMR	Frontal_Sup_Medial_R	FMOL	Frontal_Med_Orb_L
FMOR	Frontal_Med_Orb_R	RL	Rectus_L	RR	Rectus_R
FCL	FCmed_L	OFCMR	OFCmed_R	OFCAL	OFCant_L
OFCAR	OFCant_R	OFCPL	OFCpost_L	CFCR	CFCpost_R
OFCLL	OFClat_L	OFCLR	OFClat_R	IL	Insula_L
IR	Insula_R	CAL	Cingulate_Ant_L	CAR	Cingulate_Ant_R
CML	Cingulate_Mid_L	CMR	Cingulate_Mid_R	CPL	Cingulate_Post_L
CPR	Cingulate_Post_R	HL	Hippocampus_L	HR	Hippocampus_R
PHL	ParaHippocampal_L	PHR	ParaHippocampal_R	AL	Amygdala_L
AR	Amygdala_R	CAL	Calcarine_L	CAR	Calcarine_R
CL	Cuneus_L	CR	Cuneus_R	LL	Lingual_L
LR	Lingual_R	OSL	Occipital_Sup_L	OSR	Occipital_Sup_R
OML	Occipital_Mid_L	OMR	Occipital_Mid_R	OIL	Occipital_Inf_L
OIR	Occipital_Inf_R	FL	Fusiform_L	FR	Fusiform_R
POSTL	Postcentral_L	POSTR	Postcentral_R	PSL	Parietal_Sup_L
PSR	Parietal_Sup_R	PIL	Parietal_Inf_L	PIR	Parietal_Inf_R
SML	SupraMarginal_L	SMR	SupraMarginal_R	ANL	Angular_L
ANR	Angular_R	PCL	Precuneus_L	PCR	Precuneus_R
PLL	Paracentral_Lobule_L	PLR	Paracentral_Lobule_R	CAUL	Caudate_L
CAUR	Caudate_R	PUL	Putamen_L	PUR	Putamen_R
PAL	Pallidum_L	PAR	Pallidum_R	TL	Thalamus_L
TR	Thalamus_R	HEL	Heschl_L	HER	Heschl_R
TSL	Temporal_Sup_L	TSR	Temporal_Sup_R	TPSL	Temporal_Pole_Sup_L
TPSR	Temporal_Pole_Sup_R	TML	Temporal_Mid_L	TMR	Temporal_Mid_R
TPML	Temporal_Pole_Mid_L	TPMR	Temporal_Pole_Mid_R	TIL	Temporal_Inf_L
TIR	Temporal_Inf_R	CC1L	Cerebelum_Crus1_L	CC1R	Cerebelum_Crus1_R
CC2L	Cerebelum_Crus2_L	CC2R	Cerebelum_Crus2_R	C3L	Cerebelum_3_L
C3R	Cerebelum_3_R	C45L	Cerebelum_4_5_L	C45R	Cerebelum_4_5_R
C6L	Cerebelum_6_L	C6R	Cerebelum_6_R	C7L	Cerebelum_7b_L
C7R	Cerebelum_7b_R	C8L	Cerebelum_8_L	C8R	Cerebelum_8_R
C9L	Cerebelum_9_L	C9R	Cerebelum_9_R	C10L	Cerebelum_10_L
C10R	Cerebelum_10_R	V12	Vermis_1_2	V3	Vermis_3
V45	Vermis_4_5	V6	Vermis_6	V7	Vermis_7
V8	Vermis_8	V9	Vermis_9	V10	Vermis_10

Table A.6: Abbreviation codes for ROI names.