Medical/Bio Research Topics Ⅱ: Week 13 (28.11.2025)

Hands-on AI Classification Model Development (2): Model Architecture

인공지능 분류 모델 개발 실습 (2): 모델 구조

Effects of Brain Size on Sex Classification

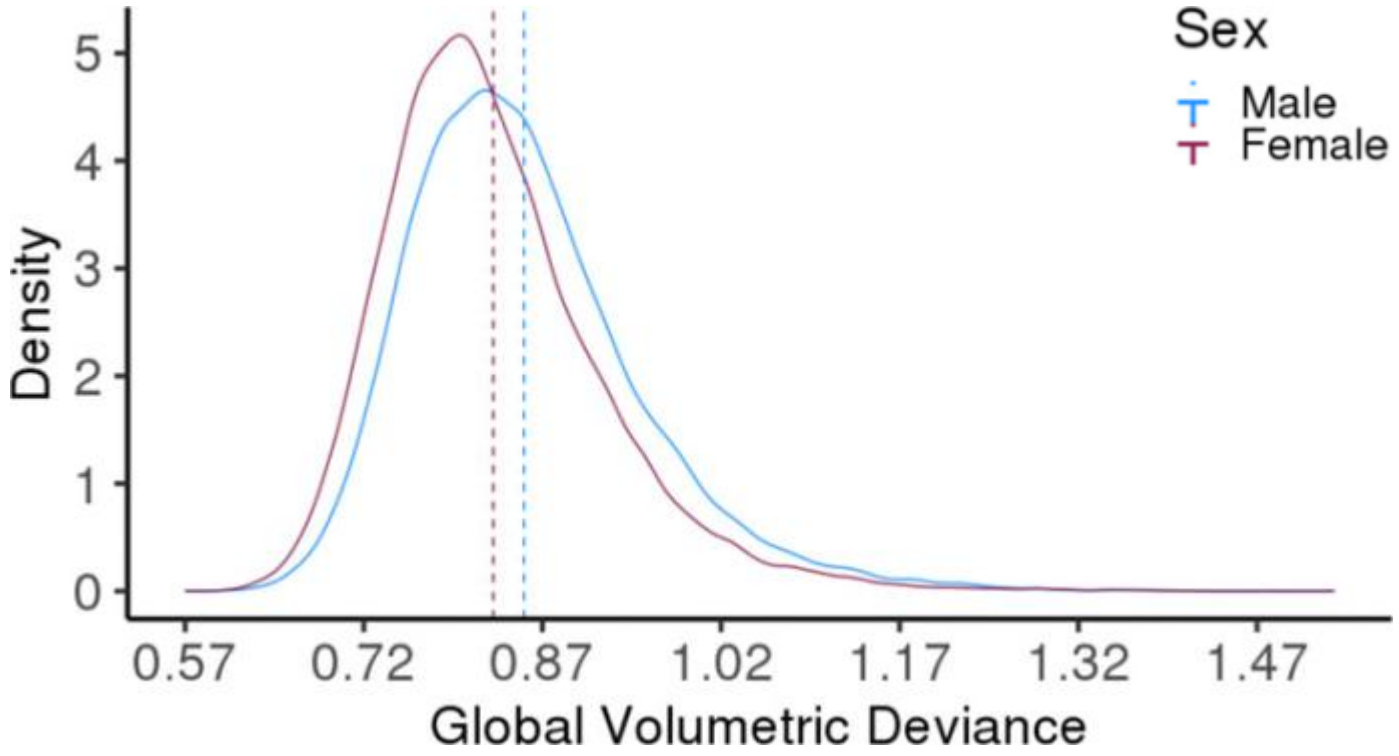
- On average, larger brain size in males than females
  - Larger total intracranial volume (TIV) in males than females by about 12% [Ruigrok et al., 2014]
  - High sex discrimination accuracy (84%) using TIV alone [Sanchis-Segura et al., 2020]
  - Sex differences not only in total brain volume but also in regional brain volumes

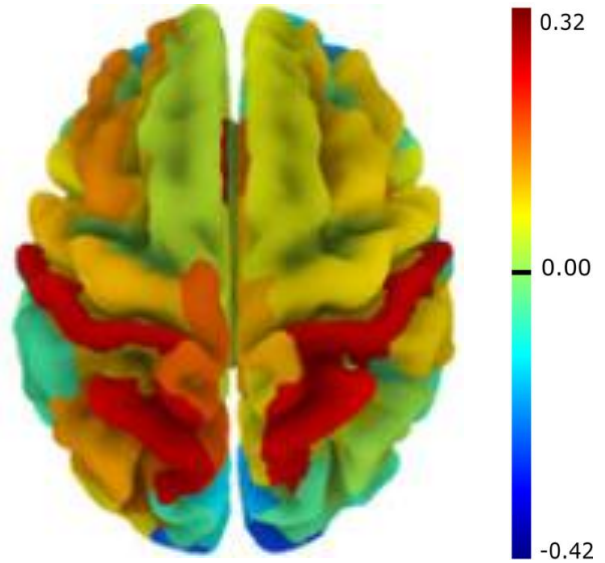
	Levene's test for equality of variances		*t*-test for equality of means					95% CI of the mean	
	F	Significance (2-tailed)	*t*	df	Significance	Mean difference	SE difference	Lower	Upper
Whole cortical volume	0.116	0.735	5.180	58	0.000	76561.23	14780.44	46974.97	106147.50
Left cortical hemisphere	0.018	0.895	5.023	58	0.000	38805.93	7725.01	23342.65	54269.22
Right cortical hemisphere	0.047	0.829	5.145	58	0.000	38320.30	7448.02	23411.47	53229.13
Left frontal cortical lobe	0.009	0.927	4.191	58	0.000	12919.23	3082.67	6748.60	19089.87
Right frontal cortical lobe	0.003	0.959	3.909	58	0.000	12004.97	3070.83	5858.04	18151.89
Left occipital cortical lobe	0.175	0.677	4.037	58	0.000	5413.17	1340.82	2729.22	8097.12
Right occipital cortical lobe	0.188	0.666	3.689	58	0.000	6106.97	1655.27	2793.59	9420.35
Left parietal cortical lobe	0.010	0.919	1.871	58	0.066	4696.90	2510.98	−329.37	9723.17
Right parietal cortical lobe	0.297	0.588	3.226	58	0.002	7875.47	2440.97	2989.33	12761.60
Left temporal cortical lobe	0.022	0.883	6.133	58	0.000	14377.30	2344.27	9684.74	19069.86
Right temporal cortical lobe	0.735	0.395	5.048	58	0.000	10571.00	2093.94	6379.53	14762.47

[Carne et al., 2006]

Sex Differences in Regional Brain Volume: Males vs. Females

- Allometric considerations in brain sex differences
  - Brain size as a confounding factor in detecting subtle brain differences between sexes
  - Even after controlling for brain size, regional patterns contribute to sex differences [Williams et al., 2021]
  - Reduction in sex classification accuracy after correcting for brain size differences [Sanchis-Segura et al., 2022]

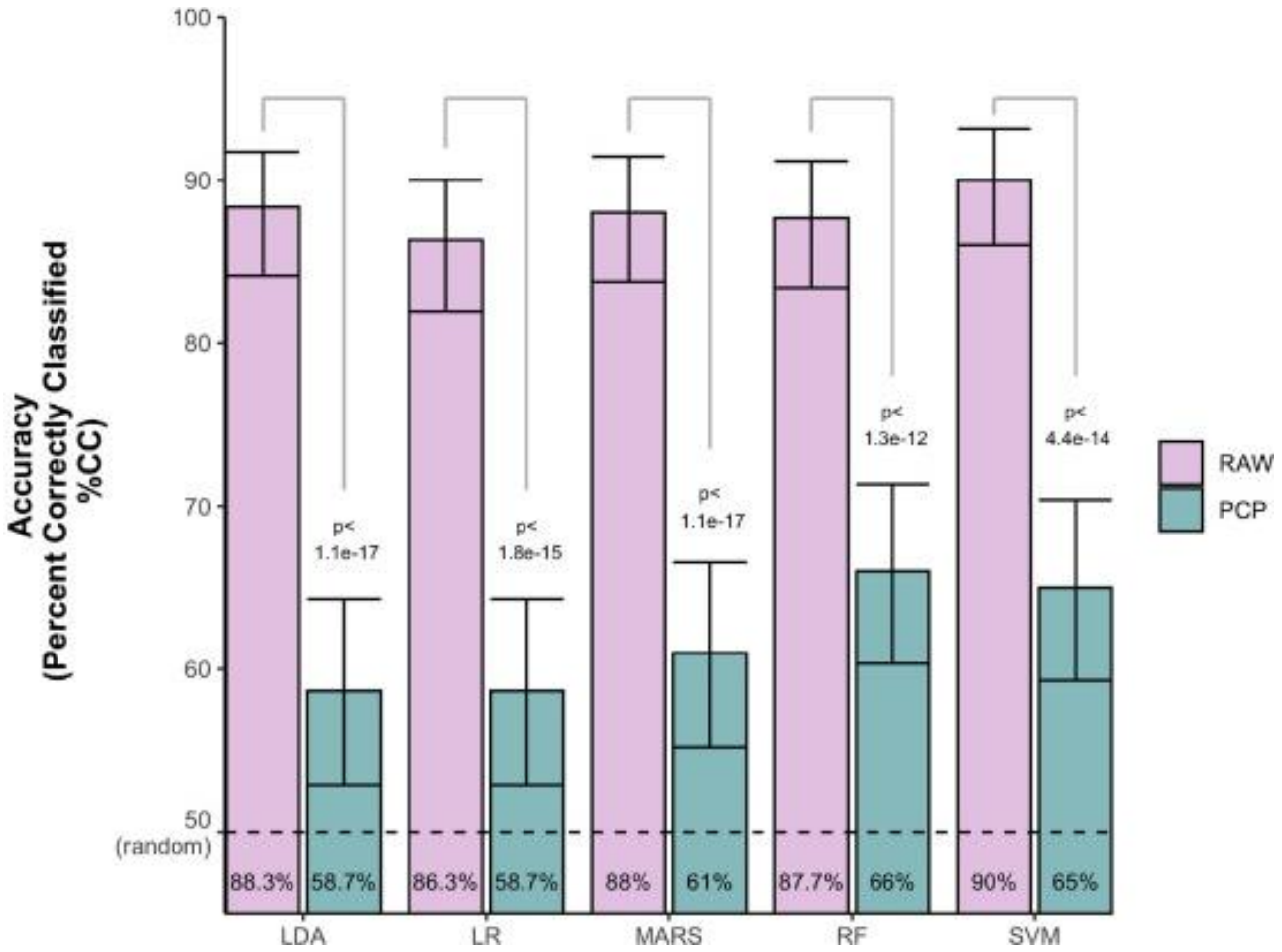




[Williams et al., 2021]

Sex Differences in Global and Cortical Volumetric Deviance: Females vs. Males

PCP, power-corrected proportion



LDA, Linear discriminant analysis
LR, Logistic regression
MARS, Multiple adaptive regression splines
RF, Random forest
SVM, Support vector machine

[Sanchis-Segura et al., 2022]

Sex Classification Accuracy without and with Correcting for Brain Size Differences

- Various brain size adjustment approaches [Sanchis-Segura et al., 2019]
  - Proportion adjustment: $\mathrm{VOI_{adj}} = \mathrm{VOI}/\mathrm{TIV}$
  - Covariate regression: $\mathrm{VOI} = \mathrm{b_0} + \mathrm{b_{TIV}}\mathrm{TIV} + \mathrm{b_{sex}}\mathrm{sex} + \varepsilon$
  - Power-corrected proportion: $\mathrm{VOI_{adj}} = \mathrm{VOI}/\mathrm{TIV^b}$
  - Residuals adjustment: $\mathrm{VOI_{adj}} = \mathrm{VOI\text{-}b}\left(\mathrm{TIV\text{-}}\overline{TIV}\right)$
  - TIV-matched subsampling: pairing each subject with the subject of the other sex with the nearest TIV
    - Only undisputed method to completely remove brain size variation

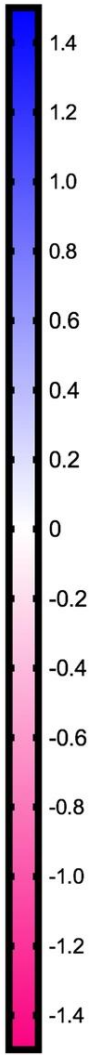
[Voevodskaya et al., 2014]

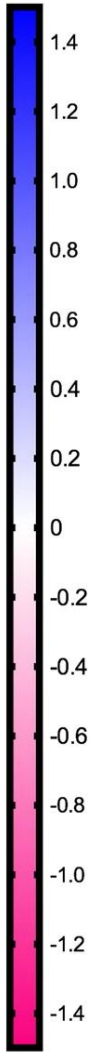
LEFT HEMISPHERE

	Raw	VBM8	Proportion	Covariate regression	PCP	Residual	TIV matched
Precentral_L	0.72		-0.24				
Frontal_Sup_L	0.66	-0.38	-0.38	-0.30	-0.27	-0.25	
Frontal_Sup_Orb_L	0.79	-0.48	-0.41	-0.40	-0.35	-0.34	-0.42
Frontal_Mid_L	0.71	-0.30	-0.24	-0.23	-0.21		
Frontal_Mid_Orb_L	0.97	-0.22		-0.22			
Frontal_Inf_Oper_L	1.06	-0.53	-0.44				
Frontal_Inf_Tri_L	0.72	-0.35	-0.32				
Frontal_Inf_Orb_L	1.09	-0.27					
Rolandic_Oper_L	1.00	-0.21					
Supp_Motor_Area_L	0.91	-0.23					
Olfactory_L	1.20						
Frontal_Sup_Medial_L	0.73	-0.29	-0.26				
Frontal_Med_Orb_L	0.70			-0.25	-0.23		
Rectus_L	0.95						
Insula_L	0.55	-0.21					
Cingulum_Ant_L	0.54	-0.24		-0.21			
Cingulum_Mid_L	0.74	-0.28					
Cingulum_Post_L	0.49						
Hippocampus_L	1.14	-0.55	-0.50				
ParaHippocampal_L	0.28						0.35
Amygdala_L	0.96			0.27	0.22	0.23	0.41
Calcarine_L	0.96						
Cuneus_L	1.15						
Lingual_L	0.68	-0.30					
Occipital_Sup_L	0.85		-0.25				
Occipital_Mid_L	0.87	-0.34	-0.26	-0.23	-0.22	-0.21	
Occipital_Inf_L	0.60						
Fusiform_L	0.57						
Postcentral_L	0.47	-0.37	-0.52	-0.27	-0.22		
Parietal_Sup_L	0.66	-0.40	-0.30				
Parietal_Inf_L	0.54	-0.59	-0.52	-0.29	-0.21	-0.21	
SupraMarginal_L	0.61	-0.42	-0.32	-0.23			
Angular_L	0.74	-0.30					
Precuneus_L	0.89	-0.39					
Paracentral_Lobule_L	0.49	-0.27	-0.27				
Caudate_L	0.55	-0.57	-0.45				
Putamen_L	1.14			0.40	0.30	0.31	0.39
Pallidum_L	0.79	-0.31		0.44	0.30	0.29	0.44
Thalamus_L	0.28	-0.70	-0.79	-0.27			-0.16
Heschl_L	0.88						
Temporal_Sup_L	0.96						
Temporal_Pole_Sup_L	0.92						
Temporal_Mid_L	0.96	-0.52	-0.27				
Temporal_Pole_Mid_L	0.93						
Temporal_Inf_L	1.15	-0.23					
Cerebelum_Crus1_L	1.12			0.31	0.23	0.25	0.34
Cerebelum_Crus2_L	0.68	-0.35	-0.30				
Cerebelum_3_L	0.49						
Cerebelum_4_5_L	0.85	-0.34					
Cerebelum_6_L	1.05	-0.33					
Cerebelum_7b_L	0.87						
Cerebelum_8_L	0.99						
Cerebelum_9_L	0.60	-0.33					
Cerebelum_10_L	0.29	-0.35	-0.46	-0.33	-0.21	-0.22	-0.45
Vermis_1_2	0.57						
Vermis_4_5	0.72	-0.40	-0.25				
Vermis_7	0.47	-0.95	-0.59				
Vermis_9	0.50	-0.37	-0.25				

RIGHT HEMISPHERE

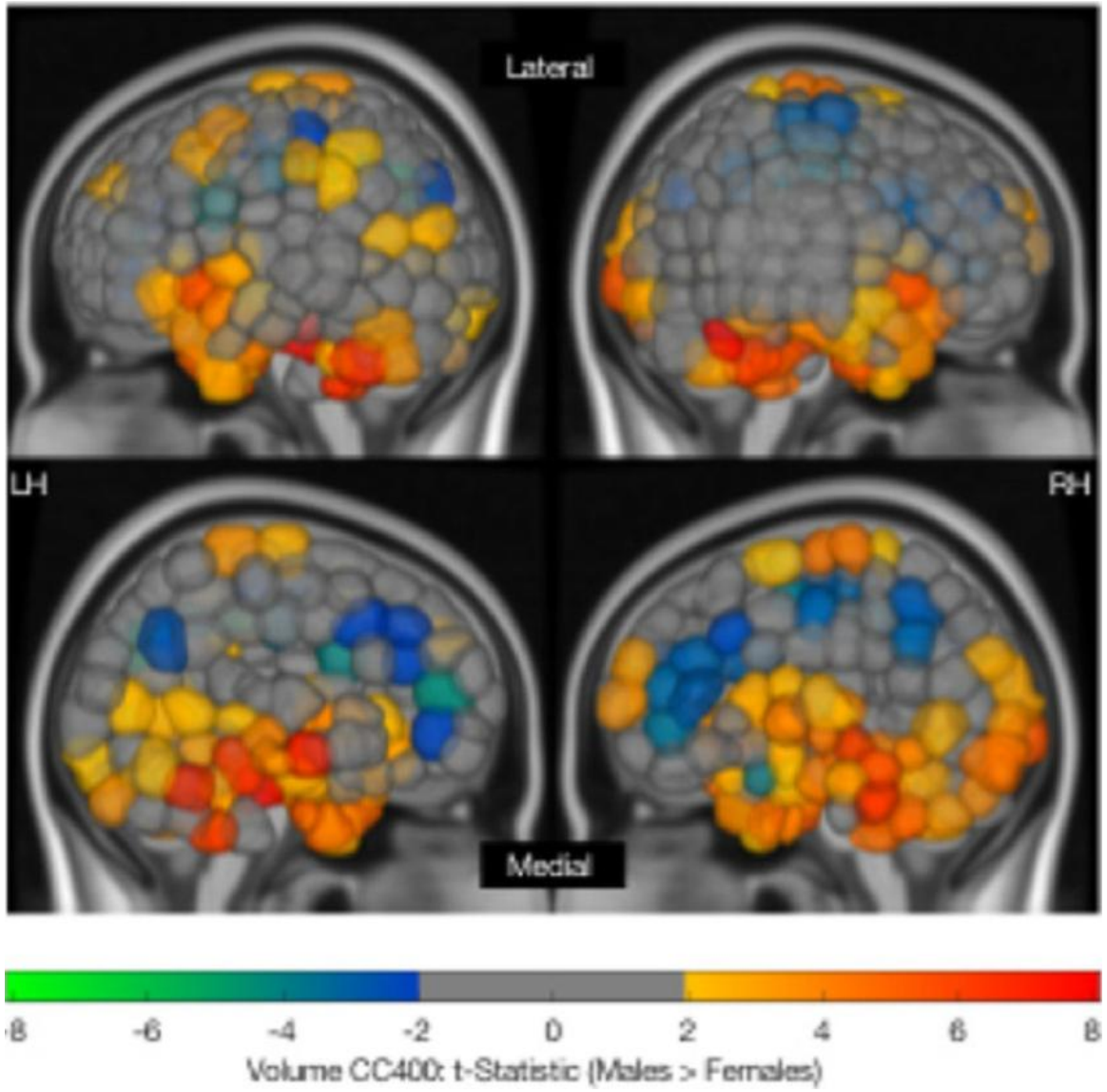
	Raw	VBM8	Proportion	Covariate regression	PCP	Residuals	TIV matched
Precentral_R	0.69	-0.32	-0.34				
Frontal_Sup_R	0.76	-0.32	-0.26				
Frontal_Sup_Orb_R	0.77	-0.44	-0.32	-0.34	-0.33	-0.31	-0.38
Frontal_Mid_R	0.87						
Frontal_Mid_Orb_R	0.85	-0.28					
Frontal_Inf_Oper_R	0.56	-0.43	-0.33				
Frontal_Inf_Tri_R	0.79	-0.30	-0.22				
Frontal_Inf_Orb_R	0.83	-0.27					
Rolandic_Oper_R	0.89						
Supp_Motor_Area_R	0.97						
Olfactory_R	0.93						
Frontal_Sup_Medial_R	0.91						
Frontal_Med_Orb_R	0.86	-0.23		-0.23	-0.24	-0.22	
Rectus_R	0.92						
Insula_R	1.09						
Cingulum_Ant_R	0.68	-0.37	-0.27	-0.22			
Cingulum_Mid_R	0.93	-0.29	-0.26				
Cingulum_Post_R	0.93						
Hippocampus_R	0.91	-0.60	-0.45				
ParaHippocampal_R	1.18						
Amygdala_R	1.39		0.30	0.30	0.29	0.29	0.61
Calcarine_R	0.70						
Cuneus_R	0.95						
Lingual_R	1.30			0.27	0.24	0.25	0.46
Occipital_Sup_R	0.55	-0.40	-0.37	-0.34	-0.27	-0.26	
Occipital_Mid_R	0.74	-0.52	-0.28	-0.29	-0.25	-0.25	-0.36
Occipital_Inf_R	0.88						
Fusiform_R	1.30		0.22	0.20			0.44
Postcentral_R	0.59	-0.44	-0.42				
Parietal_Sup_R	0.66		-0.23				
Parietal_Inf_R	0.43	-0.52	-0.52	-0.31	-0.21	-0.22	
SupraMarginal_R	0.85						
Angular_R	0.54	-0.59	-0.39	-0.25			
Precuneus_R	1.06	-0.27					
Paracentral_Lobule_R	0.55	-0.27	-0.24				
Caudate_R	0.60	-0.57	-0.40				
Putamen_R	1.09			0.34	0.26	0.26	0.34
Pallidum_R	0.83	-0.32		0.37	0.25	0.25	
Thalamus_R	0.46	-0.65	-0.67				
Heschl_R	0.69						
Temporal_Sup_R	0.78	-0.44	-0.32				
Temporal_Pole_Sup_R	0.98						
Temporal_Mid_R	0.87	-0.65	-0.37				
Temporal_Pole_Mid_R	1.10		0.24	0.29	0.21	0.23	0.33
Temporal_Inf_R	1.15	-0.37					
Cerebelum_Crus1_R	0.94	-0.23		0.28			
Cerebelum_Crus2_R	0.69	-0.44	-0.27				
Cerebelum_3_R	0.57	-0.28					
Cerebelum_4_5_R	0.97	-0.25		0.26			
Cerebelum_6_R	1.07	-0.22					0.24
Cerebelum_7b_R	0.53	-0.45	-0.38				
Cerebelum_8_R	1.00	-0.24					
Cerebelum_9_R	0.67	-0.31					
Cerebelum_10_R	0.46	-0.23	-0.31				
Vermis_3	0.45		-0.29				
Vermis_6	0.67	-0.58	-0.43				
Vermis_8	0.68	-0.49					
Vermis_10	0.50		-0.28				





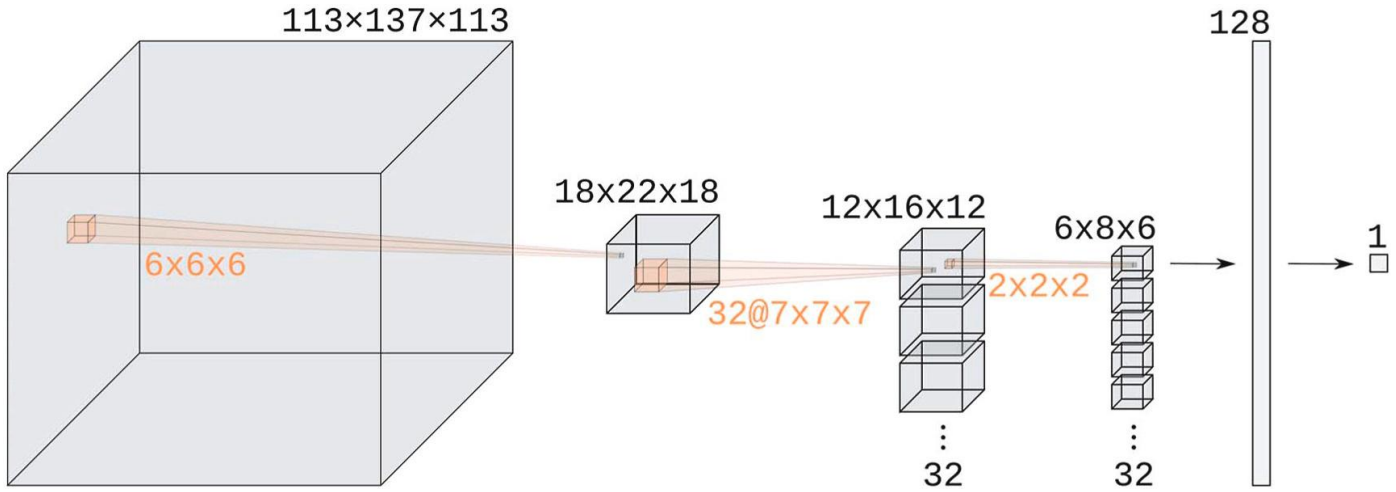
[Sanchis-Segura et al., 2019]

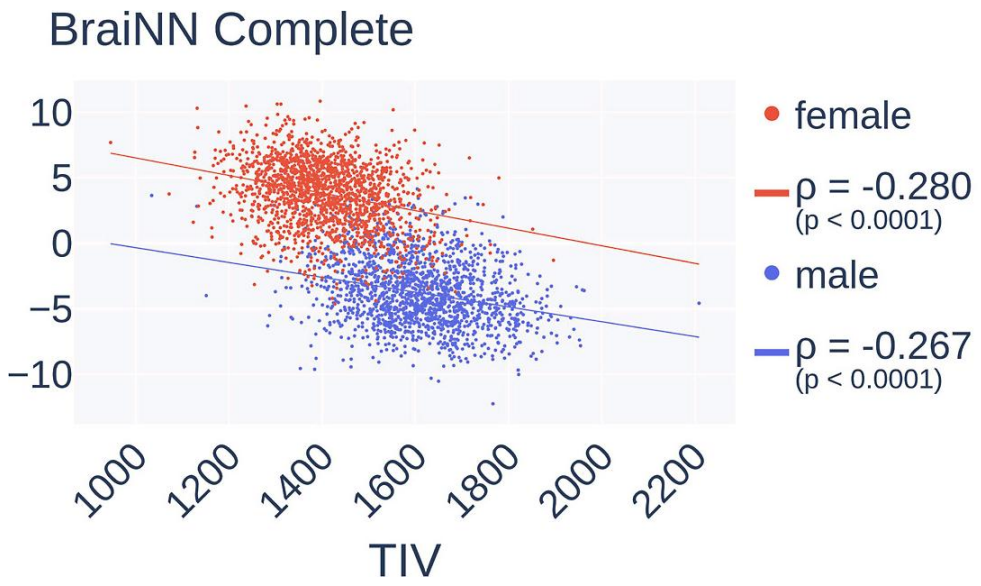
Sex Differences in Adjusted Regional Brain Volume: Males vs. Females

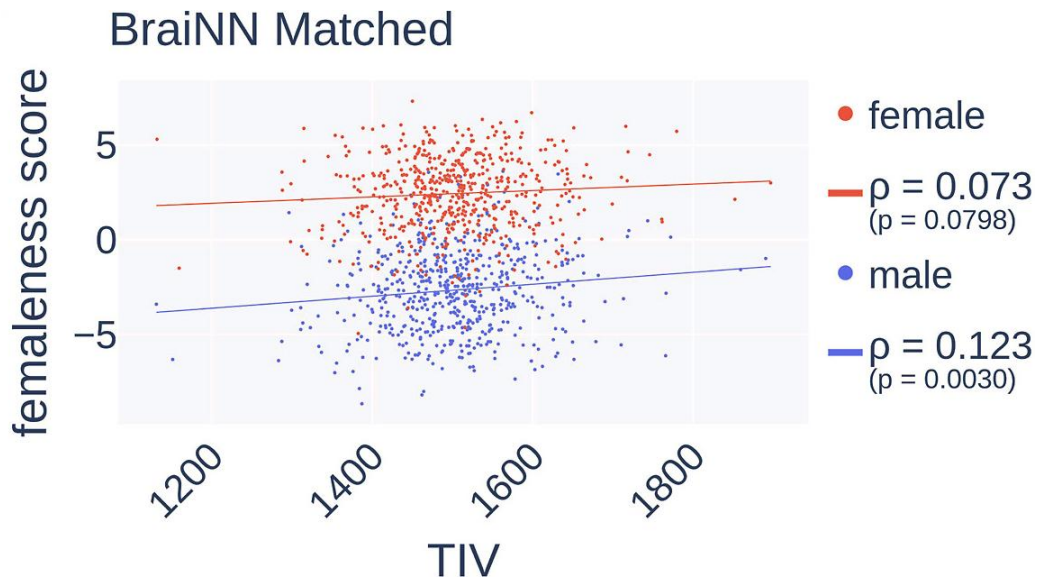


[Dhamala et al., 2020]

Sex Differences in Regional Brain Volume: Grey Matter Volume-matched Males vs. Females







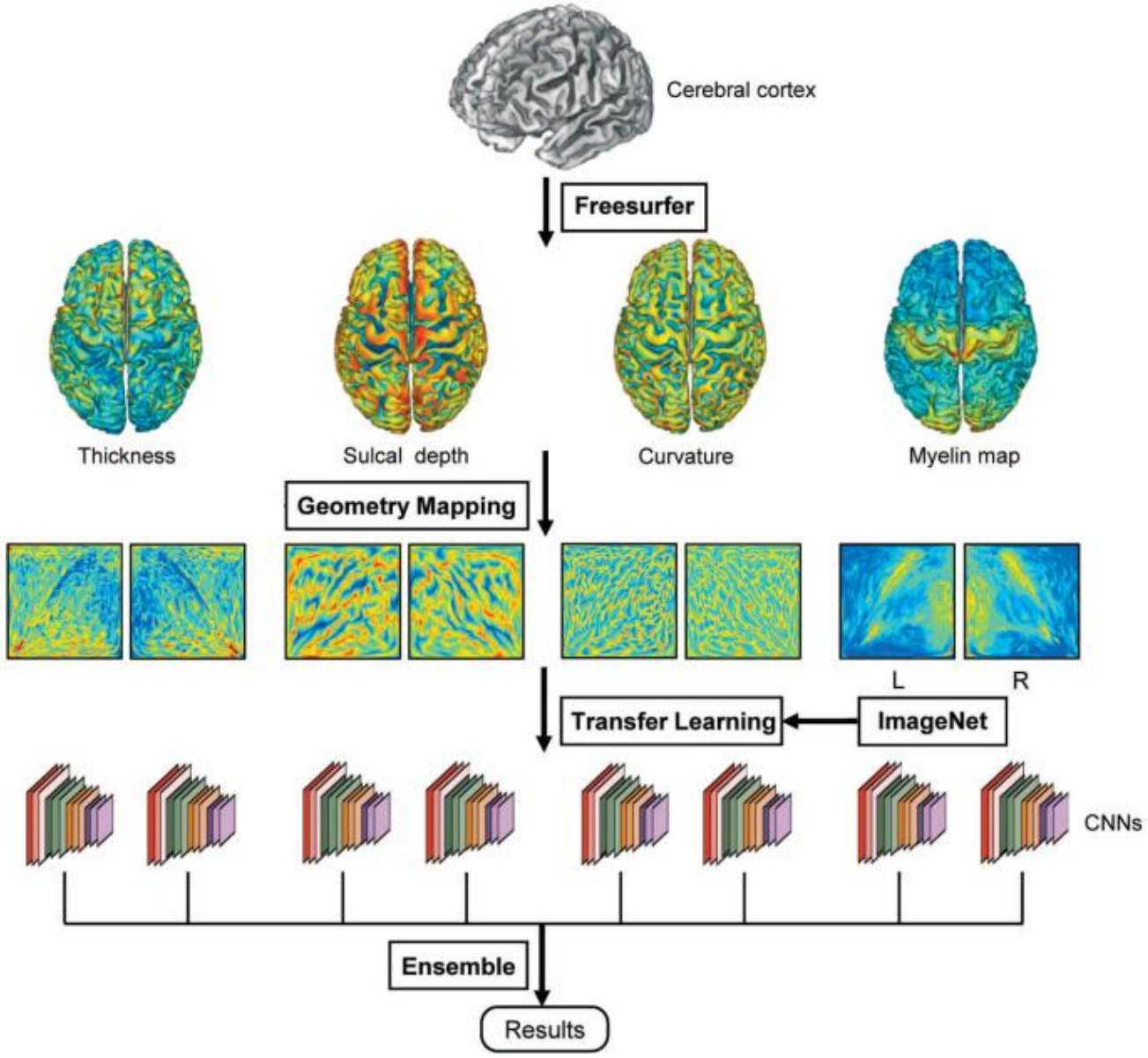
[Ebel et al., 2023]

Correlation Between Brain Sex Scores and TIV without and with TIV-matched Subsampling

Sex Classification Beyond Brain Size Effects

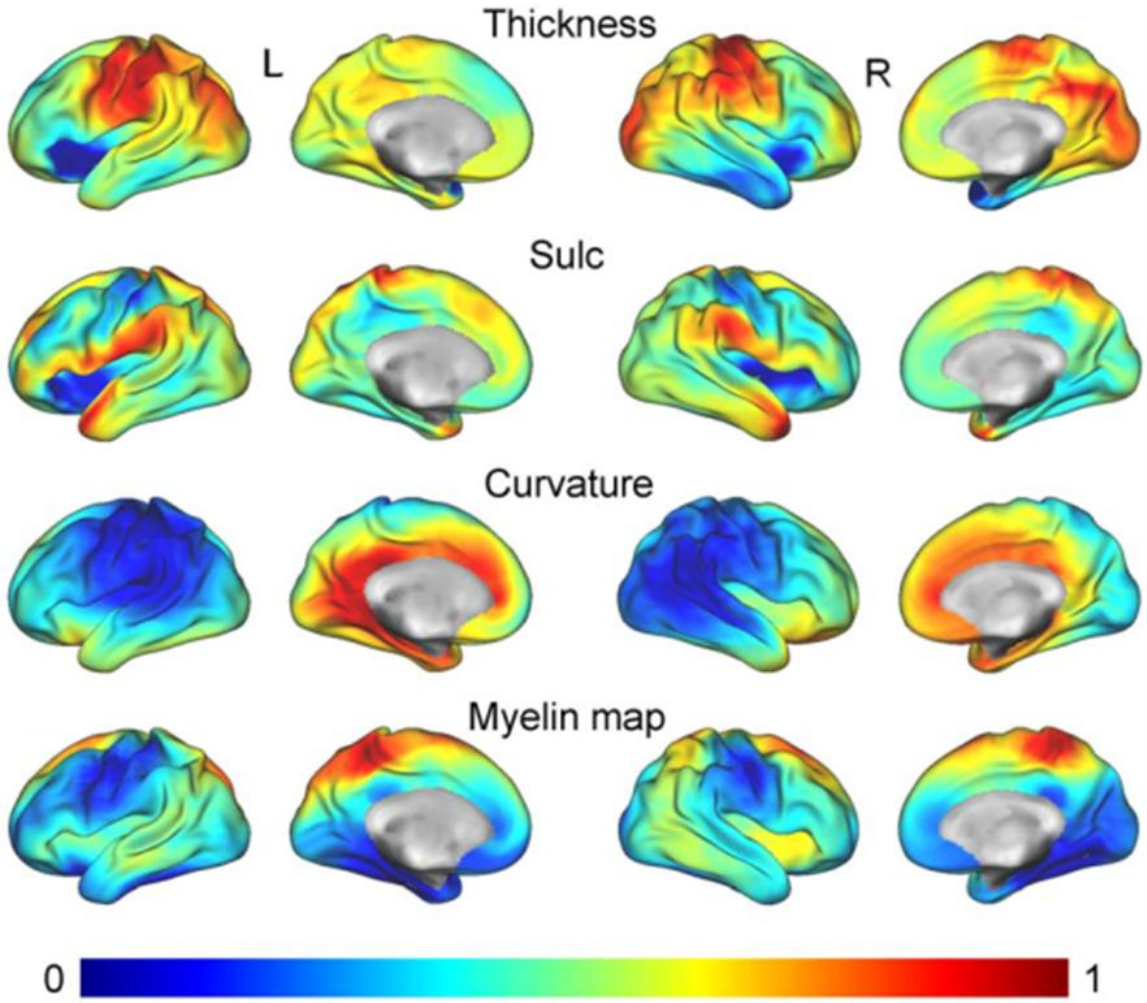
- Volume-independent approaches to sex classification
  - Surface-based cortical morphometry (structural MRI (sMRI))
    - Cortical thickness, sulcal depth, curvature, surface area
  - Functional brain dynamics (functional MRI (fMRI))
    - Temporal patterns of brain activity
  - Functional connectivity (fMRI)
    - Inter-regional correlations of brain activity
  - Structural connectivity (diffusion-weighted MRI (dMRI))
    - White matter tractography-based connectivity patterns

- Surface-based cortical morphometry [Gao et al., 2022]
  - 2D convolutional neural network (CNN): ResNet-50 and DenseNet-121 pretrained on ImageNet
  - Input: 3D cortical meshes (thickness | sulcal depth | curvature | myelin) → 2D planar (224 × 224) images
  - Datasets: n = 1,113 (505 females and 606 males)
    - 10-fold cross validation
  - Accuracy:
    - 95.06% (DenseNet)
    - 94.34% (ResNet)
  - Model explanations through occlusion sensitivity analysis with occlusion masks of a 30 × 30 black square



[Gao et al., 2022]

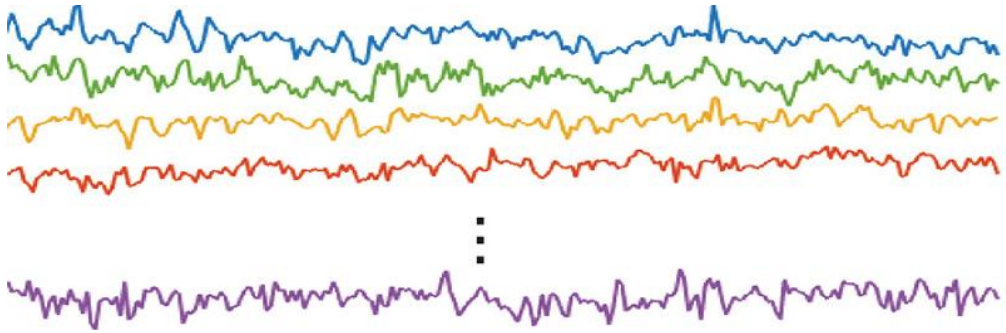
Proposed Framework Including Area-preserving Geometry Mapping for 3D Cortical Meshes



[Gao et al., 2022]

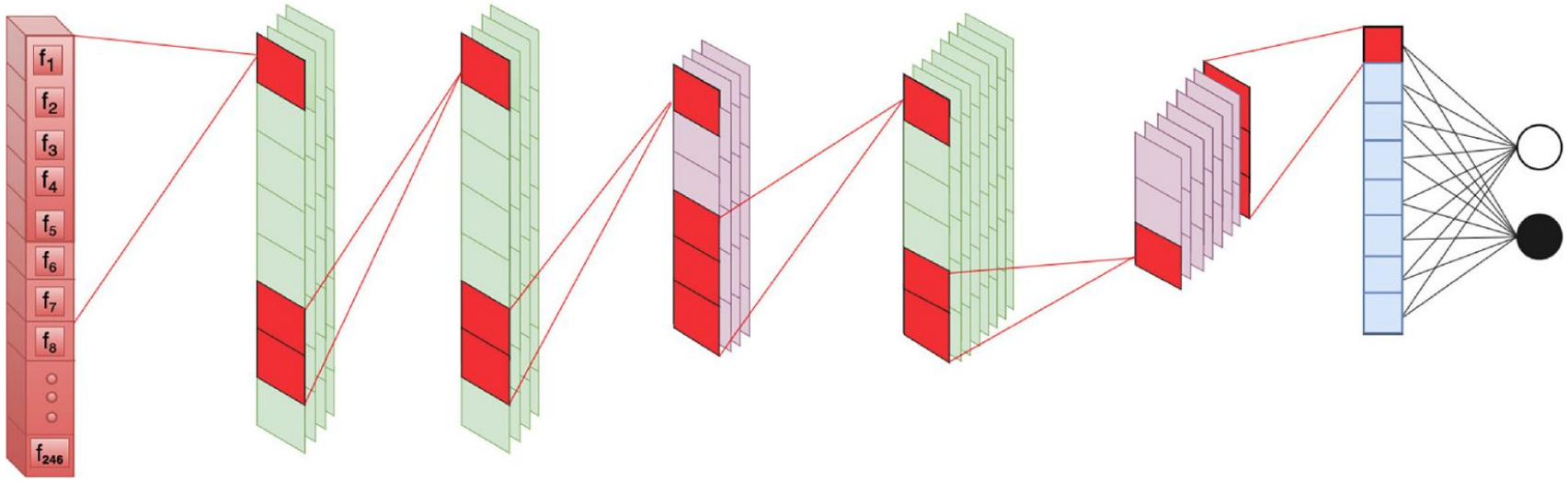
Sensitivity Maps for Sex Classification

- Functional brain dynamics [Ryali et al., 2024]
  - stDNN (Spatiotemporal Deep Neural Network)
    - Consists of two 1D CNN blocks
    - Captures latent brain dynamics features by modelling both short-term temporal correlations and long-range spatial correlations between brain regions
  - Input: Resting-state fMRI time series (246 × number of time points)
  - Datasets: *n* = 1,073 (583 females and 490 males)
    - 5-fold cross validation
    - External test 1: *n* = 205 (108 females and 97 males)
    - External test 2: *n* = 215 (78 females and 137 males)



$N_c \times N_t$

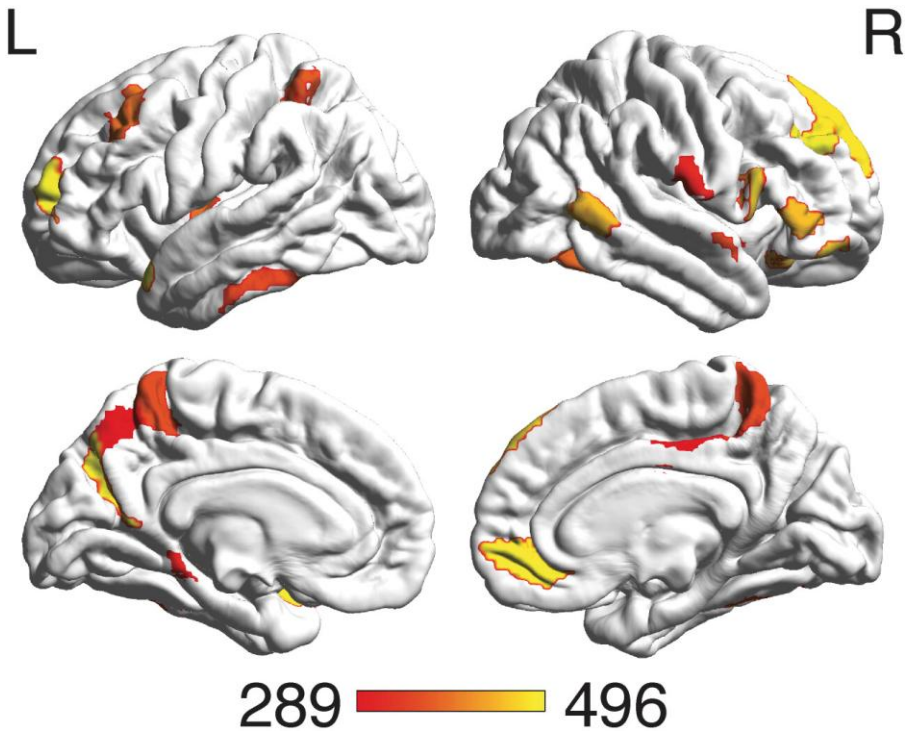
stDNN

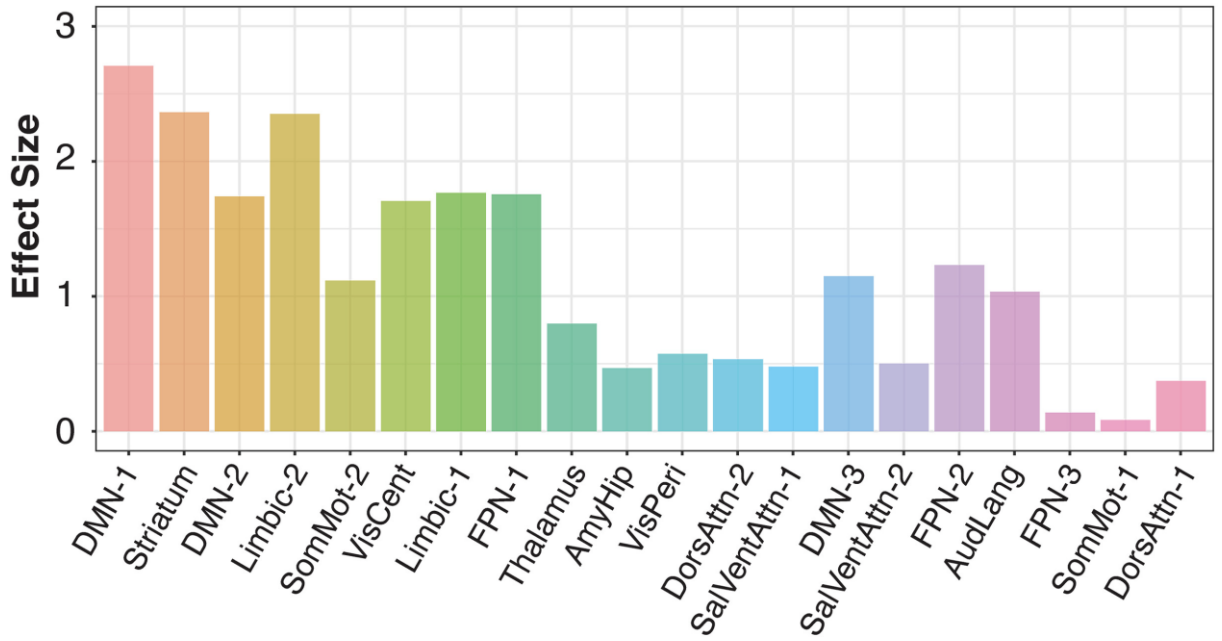


[Ryali et al., 2024]

Proposed stDNN Architecture

- Accuracy:
  - 5-fold cross validation: 90.21±1.21%
  - External test 1: 81.84±1.43%
  - External test 2: 82.60±1.68%
- Model explanations through integrated gradients-based feature attribution



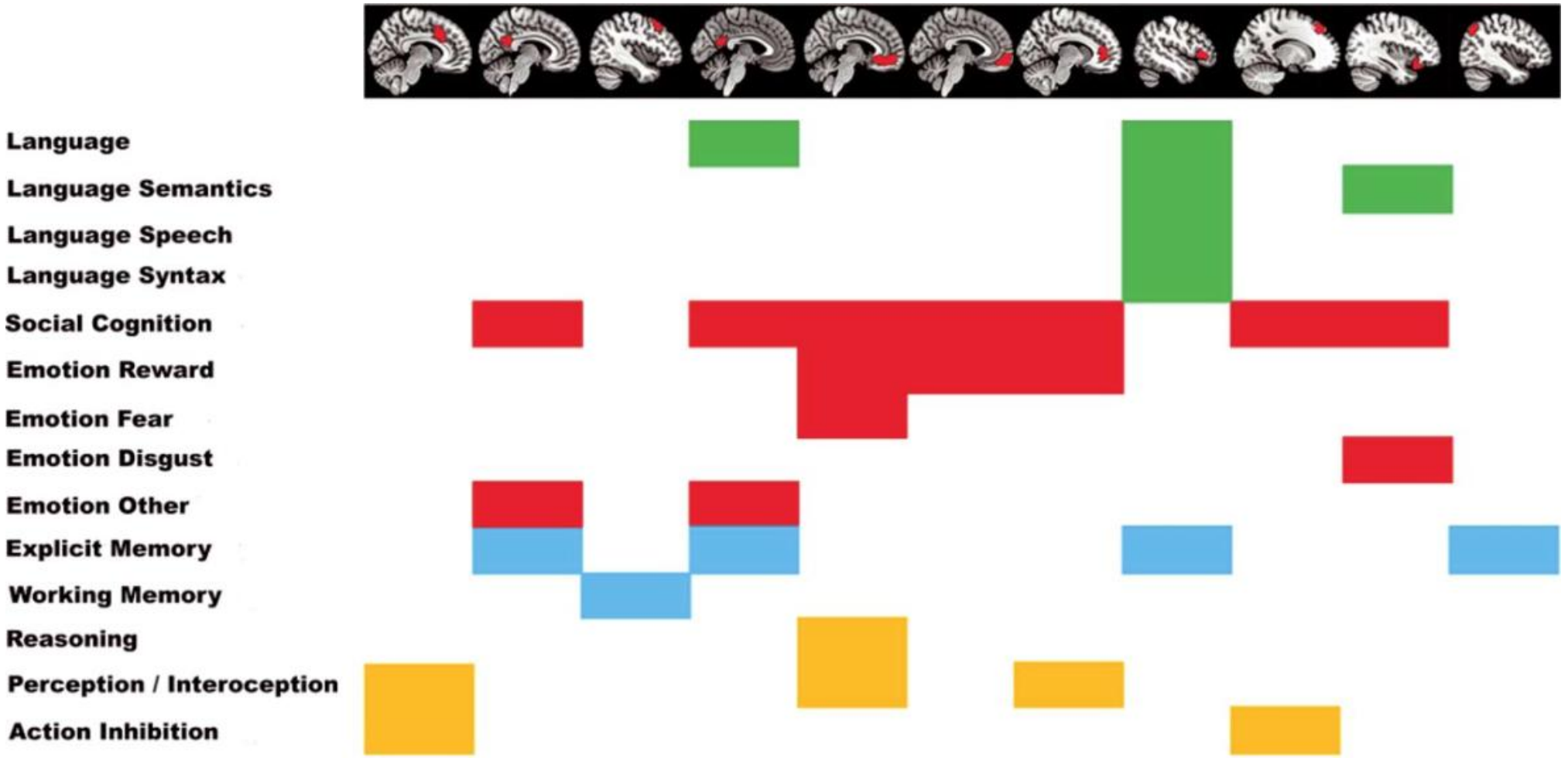


[Ryali et al., 2024l]

Consensus Maps Showing Sex-discriminating Features and Their Network-level Effect Sizes

- Functional connectivity [Weis et al., 2020]
  - Support vector machine (SVM) with radial basis function
  - Input: Resting-state functional connectivity across 436 (400 cortical (Schaefer) + 36 subcortical (Brainnetome)) brain regions
    - Between each brain region and the other brain regions
    - Between all pairs of brain regions
  - Datasets:
    - Training: *n* = 434 (217 females and 217 males)
    - Test: *n* = 310 (155 females and 155 males)
    - External test: *n* = 941 (429 females and 512 males)

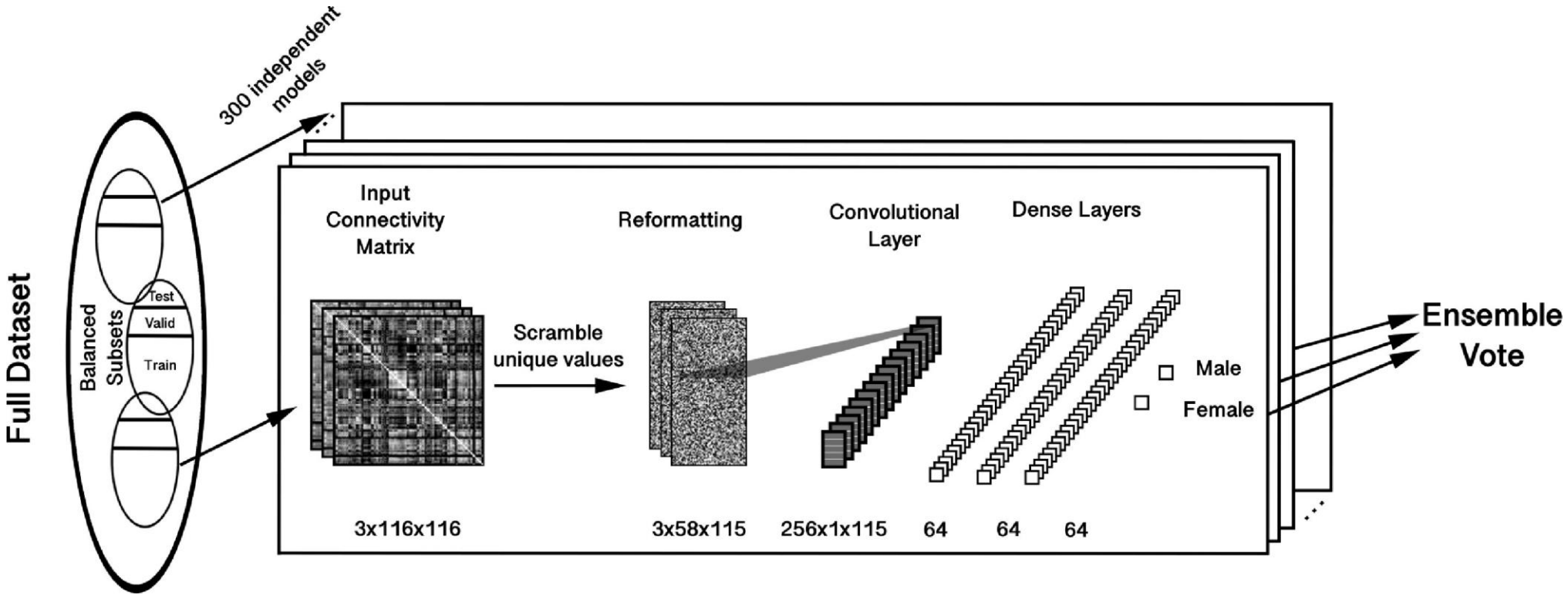
- Accuracy:
  - Functional connectivity between each region and the other regions
    - Test: 64.3±3.0% (55.4% ~ 72.6%)
    - External test: 60.0±2.5% (53.4% ~ 65.7%)
  - Functional connectivity between all pairs of regions
    - Test: 70.39%
    - External test: 51.13%



[Weis et al., 2020]

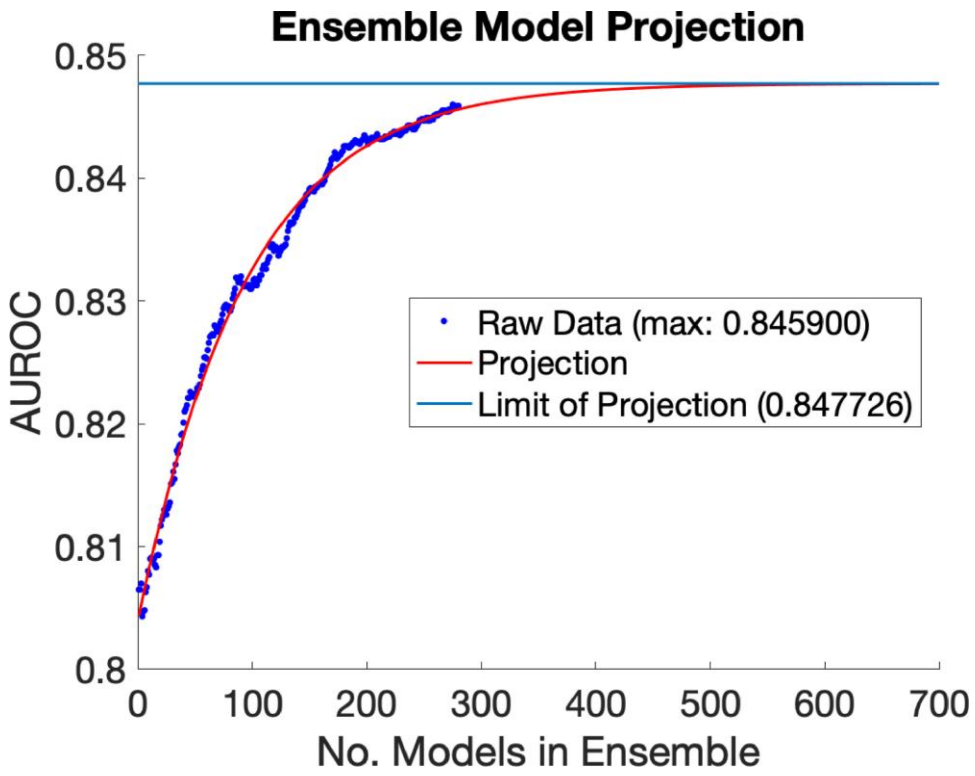
Cognitive Domains Associated with Highly Sex-predictive Brain Regions

- Functional connectivity [Leming & Suckling., 2021]
  - Stochastic encoding for an ensemble of scrambled CNNs
    - Randomly permutes the unique values of connectivity matrices to remove any spatial priors in the encoding of data
  - Input: Resting-state and task-based functional connectivity across 116 (AAL) brain regions (3 wavelet frequency bands (0.05-0.1 Hz, 0.03-0.05 Hz, and 0.01-0.03 Hz) × 116 × 116)
  - Datasets: n = 16,970
    - Training:validation:test = 4:1:1
  - Area under the receiver operating characteristic curve (AUC-ROC):
    - 0.8923 (resting-state)
    - 0.7683 (task-based)



[Leming & Suckling., 2021]

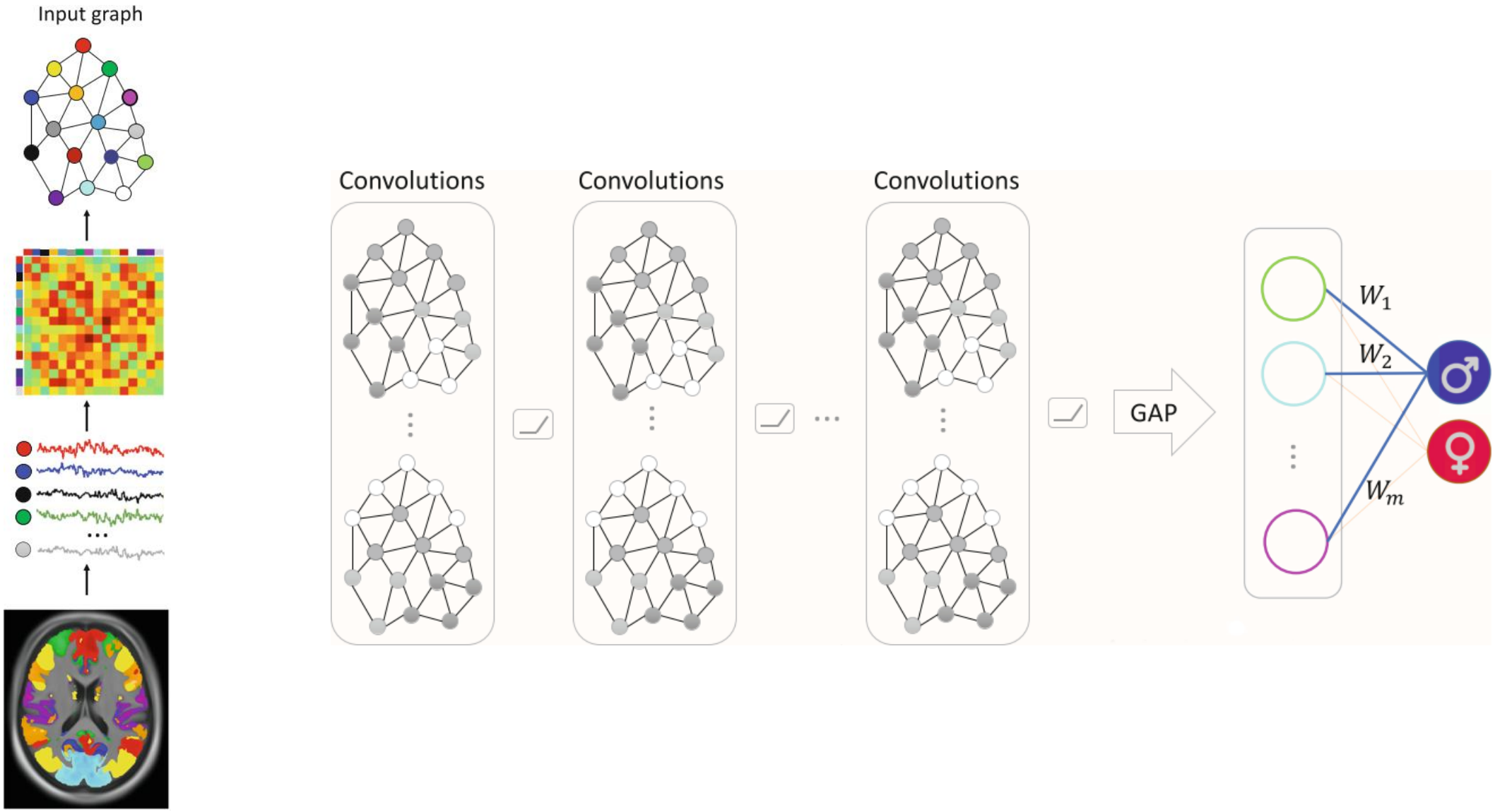
Proposed Architecture for Ensemble Learning with Scrambled CNNs



[Leming & Suckling., 2021]

Sex Classification Performance Across 1 to 300 Independent CNNs in the Ensemble Model

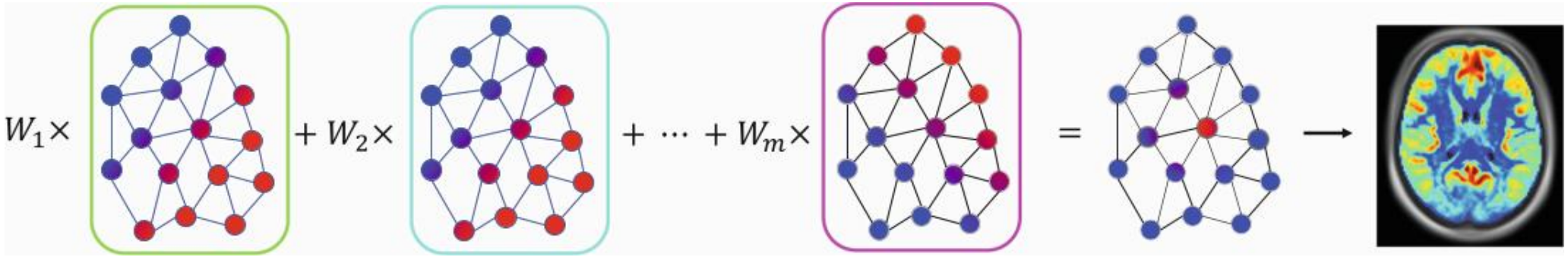
- Functional connectivity [Arslan et al., 2018]
  - ChebNet
    - Performs localized spectral graph convolutions by approximating filters as 9th-order Chebyshev polynomials of the normalized Laplacian, which is equivalent to spectral filtering but avoids explicit eigendecomposition
  - Input: Resting-state functional connectivity across 55 independent components
    - Node features: Connectivity profile
    - Edge features: Functional connectivity strength (L2-regularized partial correlation)

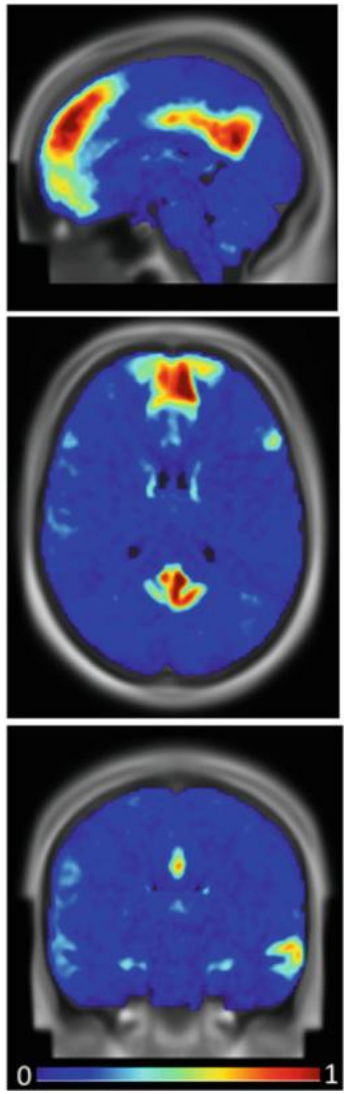


[Arslan et al.., 2018]

Proposed ChebNet Architecture

- Datasets: n = 5,430 (2,873 females and 2,557 males)
  - 10-fold cross-validation
    - Training: 80%
    - Validation: 10%
    - Test: 10%
- Accuracy:
  - 88.06±1.57%
- Model explanations through class activation mapping (CAM)



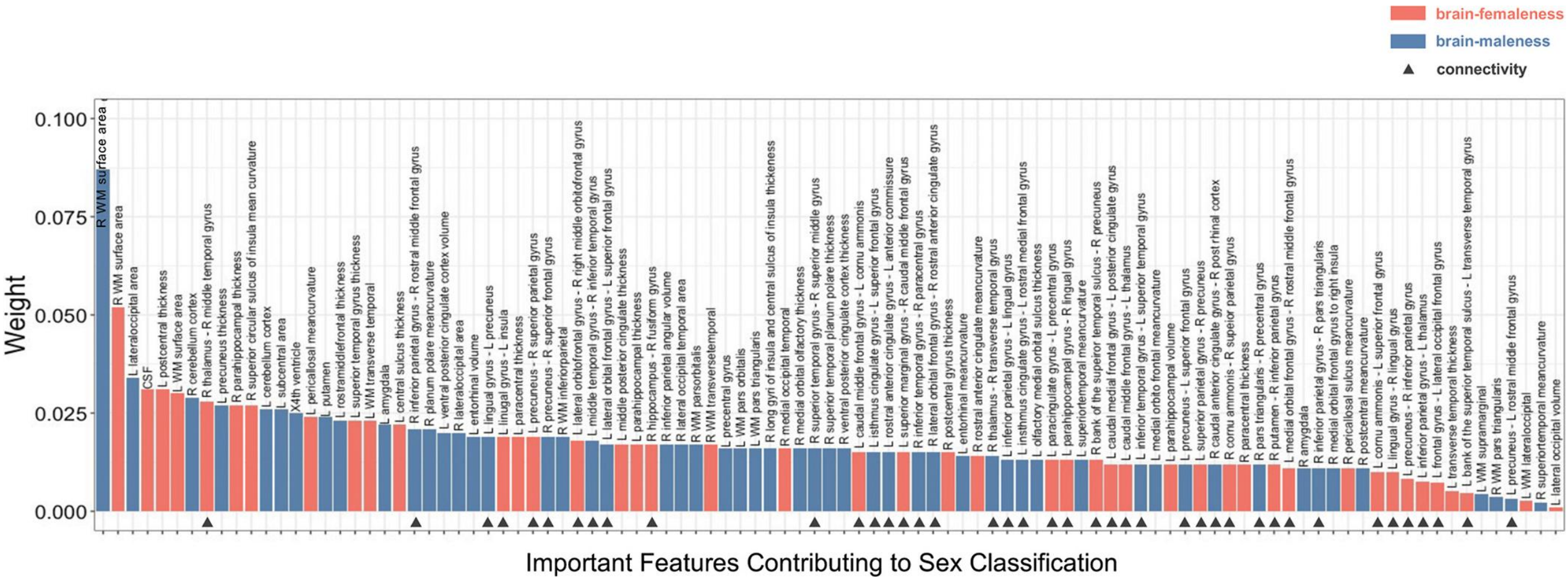


[Arslan et al.., 2018]

Saliency Maps for Sex Classification

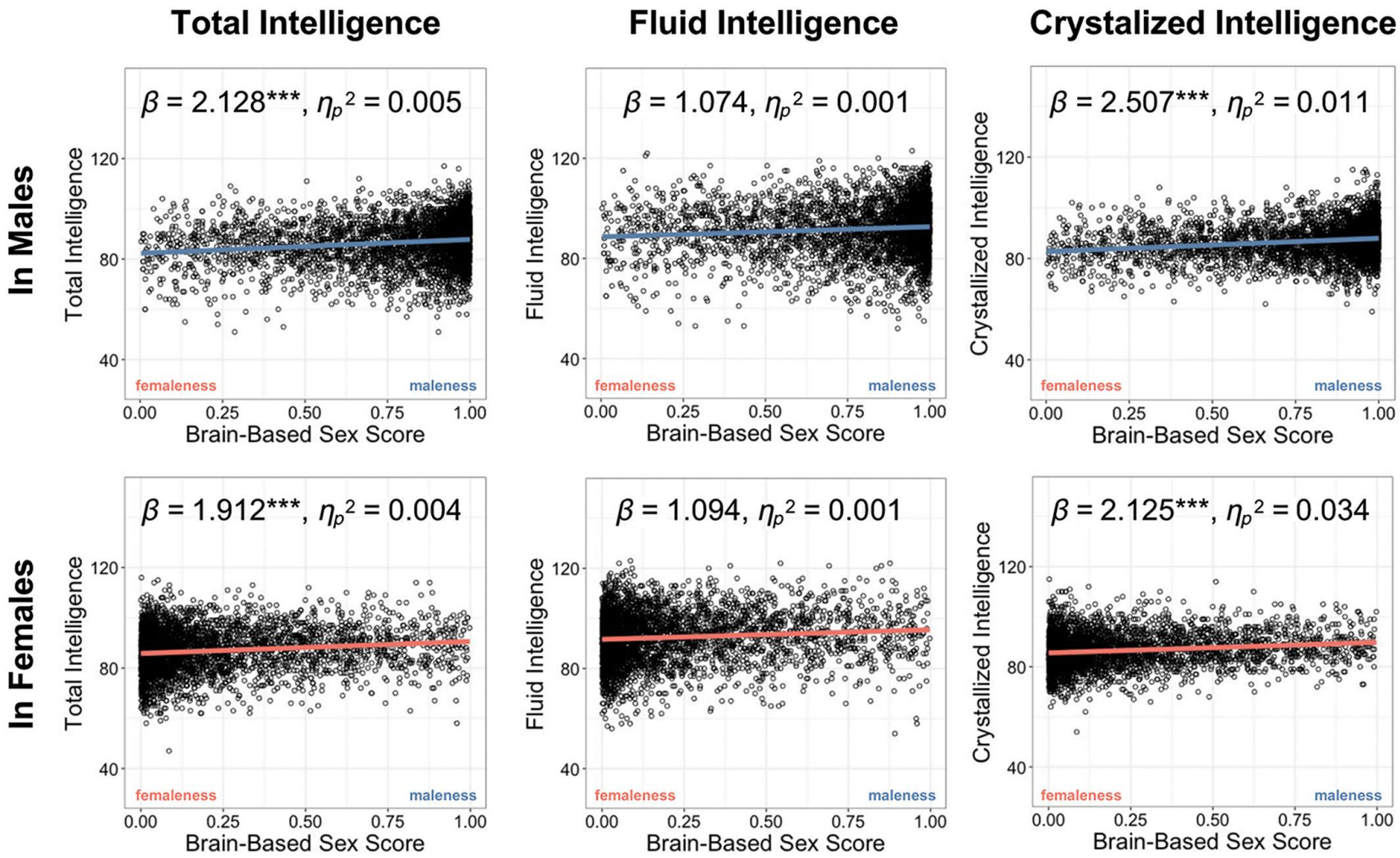
- Structural connectivity [Kim et al., 2022]
  - Ensemble of light gradient boost machine, general linear model, and XGBoost (extreme gradient boosting)
  - Input: Structural connectivity (number of streamlines) across 84 brain regions (84 × 84)
    - 3,486 features
  - Datasets: n = 9,658 (9-10 years, 4,678 females and 4,980 males)
    - Discovery (train/validation): 80%
    - Replication (test): 20%
    - Leave-one-site-out cross-validation

- AUC:
  - Discovery: 0.8505
  - Replication: 0.8501
- Model explanations through K-LIME (K-local interpretable model-agnostic explanations)



[Kim et al., 2022]

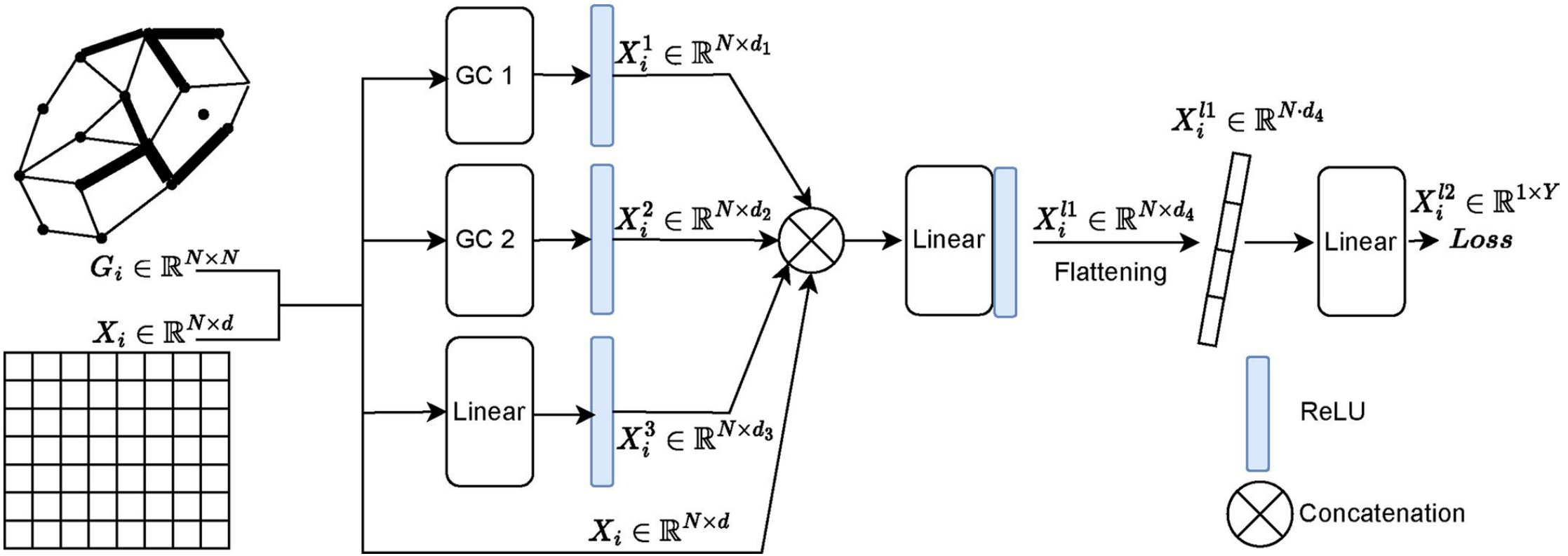
Top 100 Important Features Contributing to Sex Classification



[Kim et al., 2022]

Correlation Between Brain Sex Scores and Cognitive Intelligence

- Structural connectivity [Kazi et al., 2024]
  - Multi-head graph convolutional network (GCN)
    - Implements a multi-head architecture with four parallel branches, two of which uses the standard first-order spectral approximation to aggregate node features, while the other two are a linear (non-graph) transformation and a skip-connection branch
  - Input: structural connectivity across 85 brain regions (85×85)
    - Node features: Volume, apparent diffusion coefficient, and fractional anisotropy, connectivity profile
    - Edge features: Structural connectivity strength (number of streamlines)



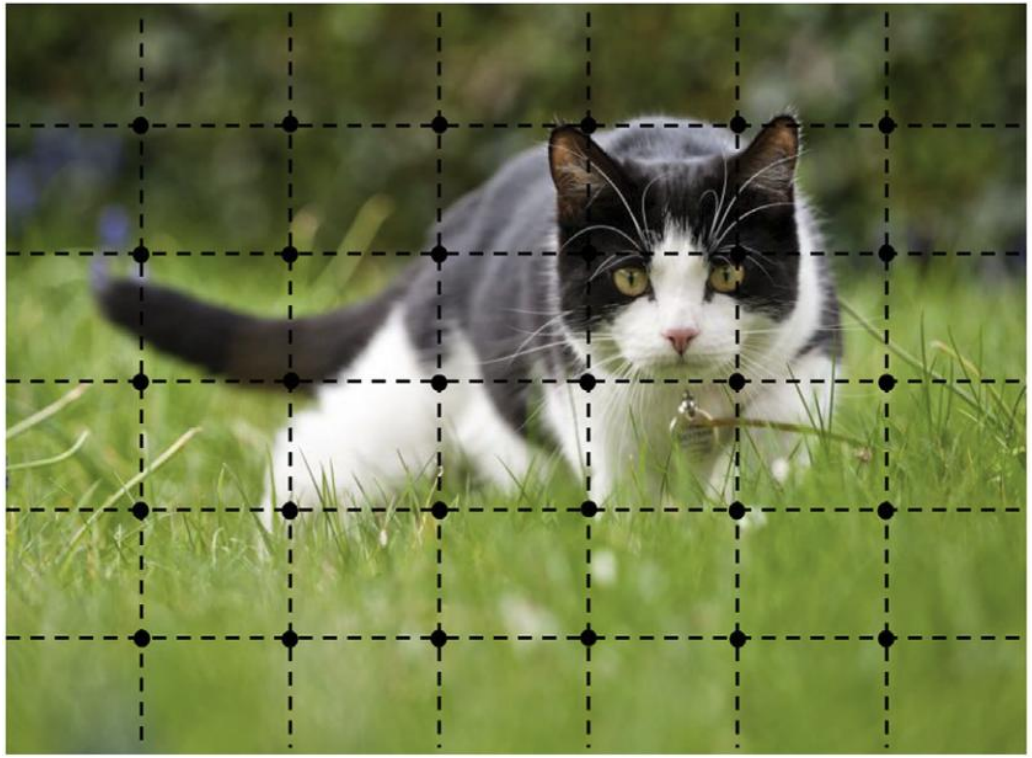
[Kazi et al., 2024]

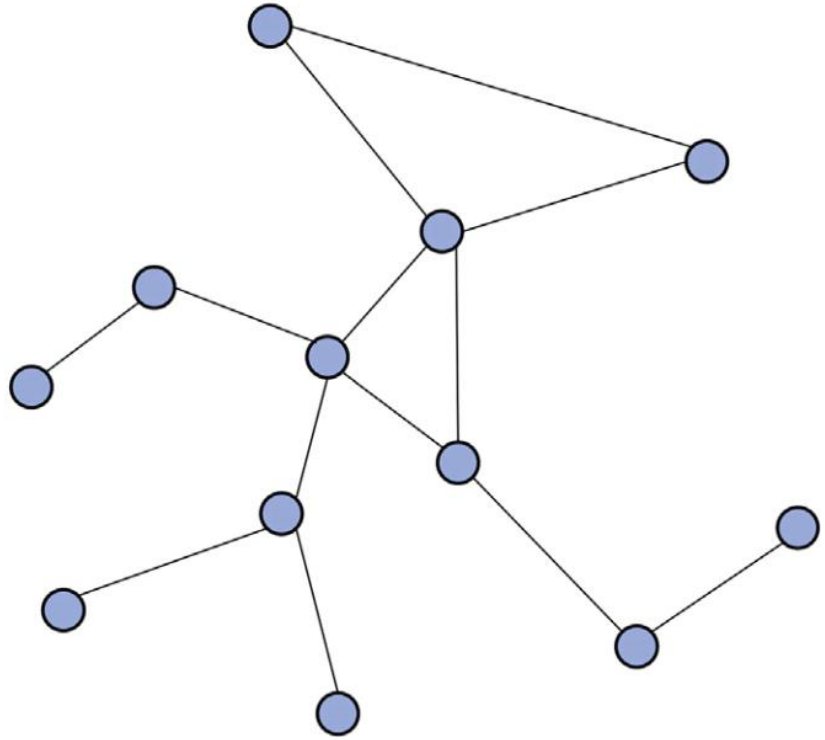
Proposed GCN Architecture

- Datasets: n_1 = 789 (511 females and 199 males) and n_2 = 1,294 (649 females and 515 males)
  - Training and validation: 90%
    - 10-fold cross-validation
  - Test: 10%
- Accuracy:
  - Validation:
    - 90.6±6.8%, 88.6±4.1% (with augmentation)
    - 89.5±6.1%, 87.8±6.6% (without augmentation)
  - Test:
    - 78.5%, 95.0%

Graph Neural Network (GNN)

- Neural network designed to process and learn from data represented in non-Euclidean domains, specifically graph-structured data
  - Non-Euclidean domain: Any geometric space, such as graphs/networks and spherical and hyperbolic manifolds, that does not follow standard Euclidean geometric rules
- Capable of learning from both node features and graph structure

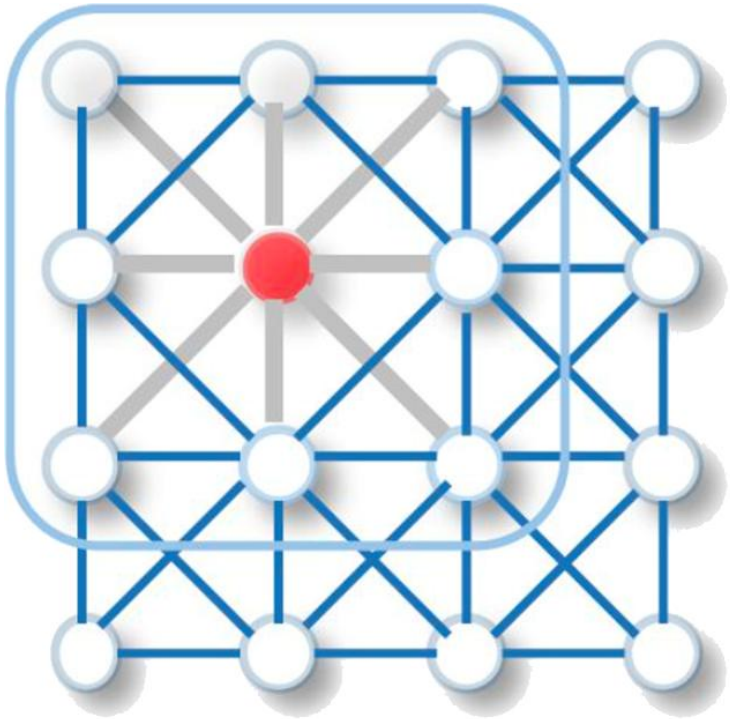




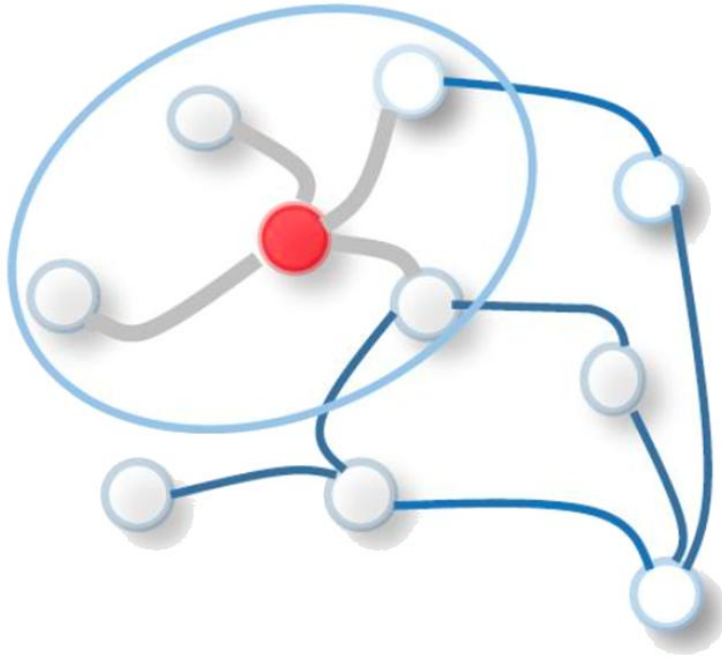
[Zhou et al., 2020]

Image in Euclidean Space vs. Graph in Non-Euclidean Space

- Generalization of CNNs to non-Euclidean domains
  - Learns hierarchical features from local structures while maintaining equivariance
    - CNNs: Spatial equivariance, GNNs: Permutation equivariance
  - Spatial GNNs
    - Direct analogy to CNNs
    - Message passing / neighborhood aggregation ≈ CNN's convolution operation
  - Spectral GNNs
    - Less visually analogous to CNNs but mathematically equivalent in terms of filtering operations in transformed domains: Input * Filter = IFFT(FFT(Input) × FFT(Filter))



2D-Convolution



Graph Convolution

[Wu et al., 2021]

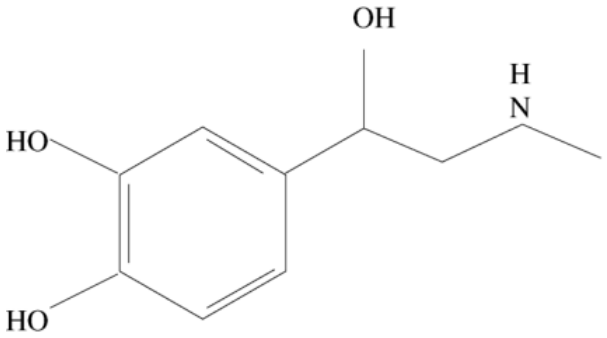
CNN vs. GNN

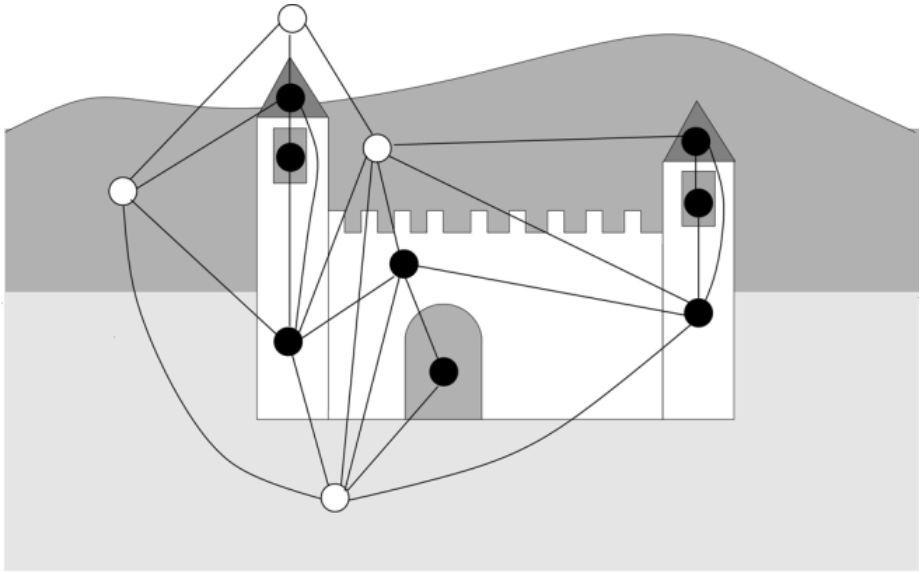
- Data representation using graph structures
  - Basic graph representation components
    - Nodes: Individual entities
    - Edges: Relationships/connections between entities
    - Node attributes
    - Edge attributes
    - Structure:
      - How nodes are connected
      - How edges are formed
      - How attributes are assigned
  - Relationships between entities are as important as the entities themselves

- Common graph-structured data
  - Molecules or chemical compounds
    - Nodes: Atoms
    - Edges: Chemical bonds
  - Images
    - Nodes: Pixels, super-pixels, or image regions
    - Edges: Spatial adjacency or feature similarity
  - World Wide Web
    - Nodes: Web pages, websites, or domains
    - Edges: Hyperlinks, references, or domain relationships

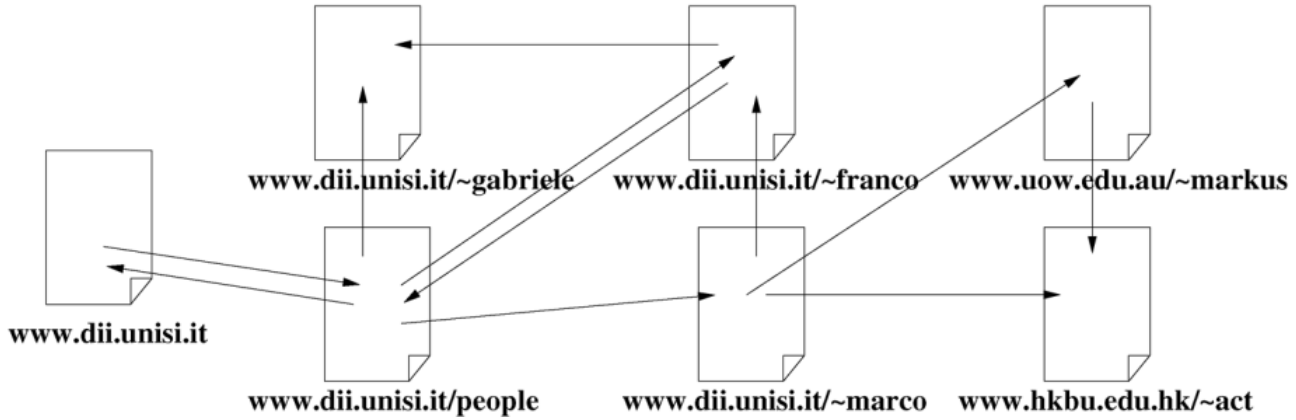
Image

Chemical compound (adrenaline)





Subset of the web



[Scarselli et al., 2009]

Data Representation Using Graph Structures

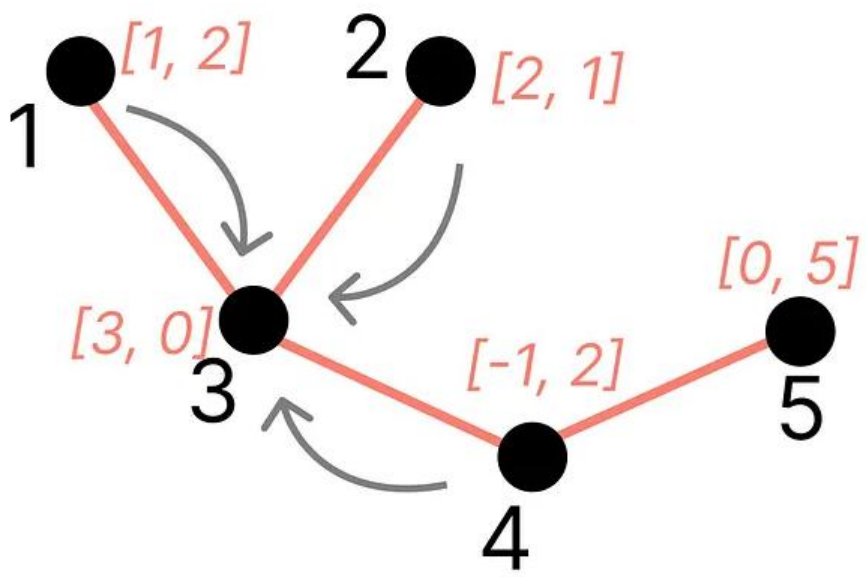
- Connections between traditional graph concepts and GNNs
  - Graph representation
    - Traditional: Mathematical structure made of nodes and edges
    - GNN: Learnable representations of nodes and edges
  - Nodes
    - Traditional: Discrete points in graph
    - GNN: Feature vectors/embeddings for each node
  - Edges
    - Traditional: Connections between nodes
    - GNN: Message paths (node-centric) and learnable edge features (edge-aware)

- Neighborhood structure
  - Traditional: Adjacent nodes defined by edges
  - GNN: Information aggregation from neighboring nodes
- Path and distance
  - Traditional: Sequence of edges between nodes
  - GNN: Message passing across multiple layers
- Centrality and importance
  - Traditional: Various centrality measures
  - GNN: Learned importance through attention mechanisms

Distinctions between GNNs and CNNs

- Data structure
  - GNNs:
    - Operate on irregular graph structures
    - Variable neighborhood size
  - CNNs:
    - Operate on regular grid-like structures
    - Fixed neighborhood size

- Operations
  - GNNs:
    - Message passing between nodes through edges
    - Aggregation functions for variable-sized neighborhoods
  - CNNs:
    - Fixed-size kernels applied to regular grids
    - Pooling operations on regular grids
- Invariance properties
  - GNNs:
    - Permutation invariant to node ordering
  - CNNs:
    - Translation invariant



[1, 2]
+ [2, 1]
+ [-1, 2]

[2, 5] ← Message for node 3

[2, 5]+[3, 0]=[5, 5]
New feature vector for node 3

Step 1: Message
m_ij = MESSAGE(h_i, h_j, e_ij)
edge features ([1,2], [2,1], [-1,2])

Step 2: Aggregate
m_i = AGGREGATE({m_ij : j ∈ N(i)})
[1,2]+[2,1]+[-1,2] = [2,5]

Step 3: Update
h_i_new = UPDATE(h_i, m_i)
[3,0] + [2,5] = [5,5]

[https://medium.com/@volzhinnv/graph-isomorphism-networks-where-gnns-become-injective-030eeb0cbc37]

Message Passing Schematic

- Localization
  - GNNs:
    - Variable receptive fields based on graph structure
  - CNNs:
    - Fixed geometric receptive fields
- Computational considerations
  - GNNs:
    - Computational complexity proportional to graph structure
  - CNNs:
    - Fixed computation based on input size

Aspect	GNN	CNN	Difference
Data structure	Graph (irregular, non-Euclidean)	Grid (regular, Euclidean)	Topology
Basic unit	Node i in graph	Pixel/location at (h, w)	Where data resides
Spatial domain	*N* nodes (arbitrary connectivity)	2D grid: $H \times W$ positions	CNN: ordered; GNN: unordered
Feature representation	*F* features per node	*C* channels per location	What describes each unit
Input shape	[Nodes, Features] + Edge_index	[Batch, Channels, Height, Width]	Batch handling
Neighborhood	Variable degree k-hop neighbors	Fixed receptive field	CNN: local grid; GNN: graph structure

GNN vs. CNN: Structural Correspondence

Operation	GNN	CNN	Purpose
Core layer	Graph convolutional layer	Convolutional layer	Feature extraction
Aggregation	Message passing = aggregation from neighbors	Convolution = weighted sum of neighbors	Information flow
Weight sharing	Same message function for all edges	Same filter across all locations	Parameter efficiency
Nonlinearity	ReLU, BatchNorm after graph conv	ReLU, BatchNorm after conv	Expressiveness
Dimension evolution	Nodes→ (constant), Features↑	Spatial↓, Channels↑ (progressive)	Abstraction

GNN vs. CNN: Operations and Transformations

GNN architecture

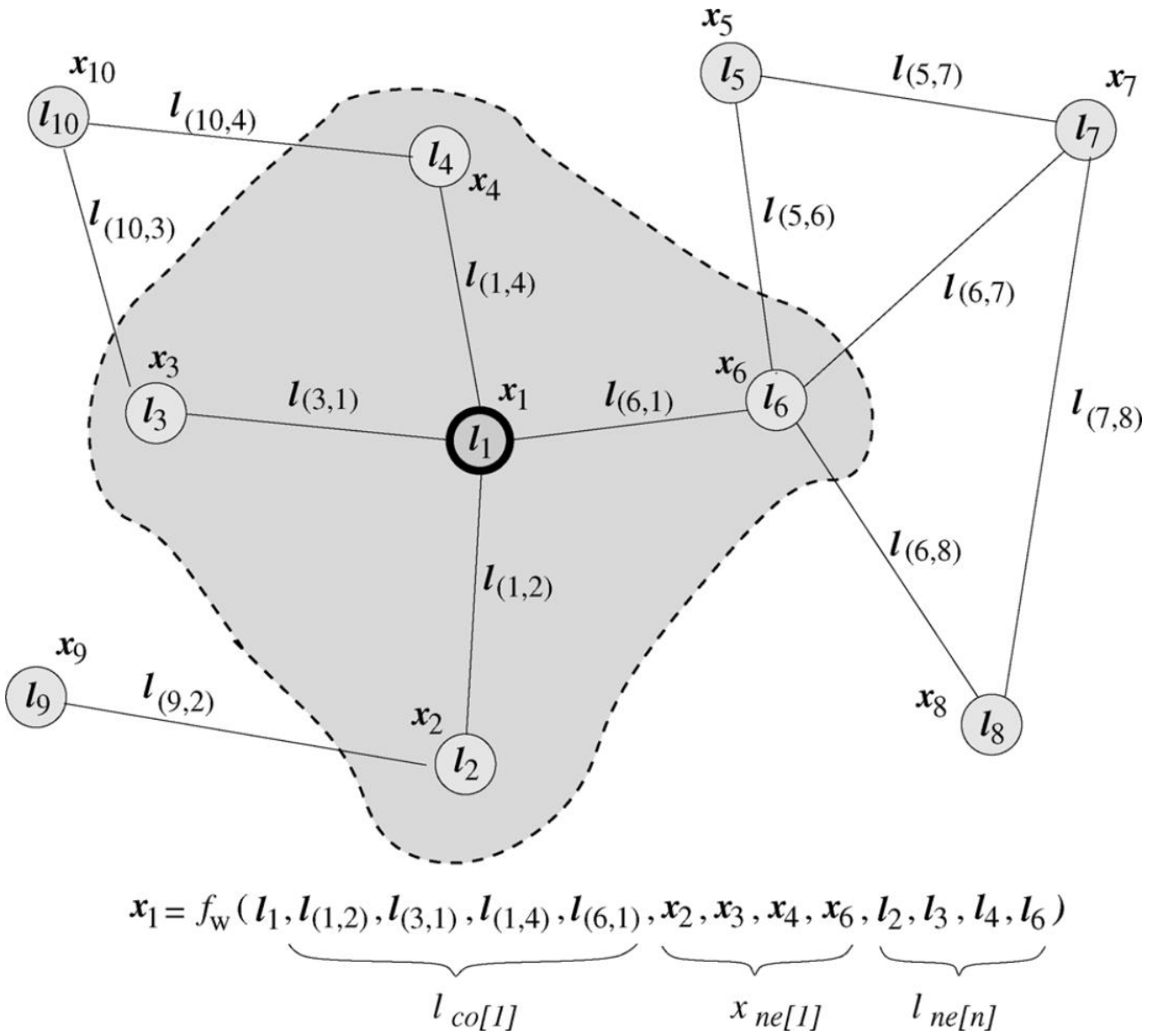
Input Graph: [N, 3] # 246 nodes, 3 features
↓ GCNConv(3, 64)
[N, 64] # Nodes: 246, Features: ↑ 64
↓ GCNConv(64, 128)
[N, 128] # Nodes: 246, Features: ↑ 128
↓ GCNConv(128, 256)
[N, 256] # Nodes: 246, Features: ↑ 256
↓ global_mean_pool
[1, 256] # Nodes: ↓ 1, Features: 256
↓ FC
[num_classes]

CNN architecture

Input Image: [B, 3, 224, 224]
↓ Conv2d(3, 64)
[B, 64, 224, 224] # Spatial: 224×224, Channels: 64
↓ MaxPool2d(2)
[B, 64, 112, 112] # Spatial: ↓ 112×112, Channels: 64
↓ Conv2d(64, 128)
[B, 128, 112, 112] # Spatial: 112×112, Channels: ↑ 128
↓ MaxPool2d(2)
[B, 128, 56, 56] # Spatial: ↓ 56×56, Channels: 128
↓ Conv2d(128, 256)
[B, 256, 56, 56] # Spatial: 56×56, Channels: ↑ 256
↓ AdaptiveAvgPool2d((1, 1))
[B, 256, 1, 1] # Spatial: ↓ 1×1, Channels: 256
↓ Flatten + FC
[B, num_classes]

Development of GNNs

- Early GNNs
  - "The Graph Neural Network Model" [Scarselli et al., 2009]
    - Establishes the first formal framework for neural networks on graphs
    - Proposes recursive approach for node representation learning (recursive node state updates based on neighborhood information until convergence)
    - Lays the groundwork for the more generalized message passing neural network (MPNN) framework
  - Feature handling: Both node and edge features



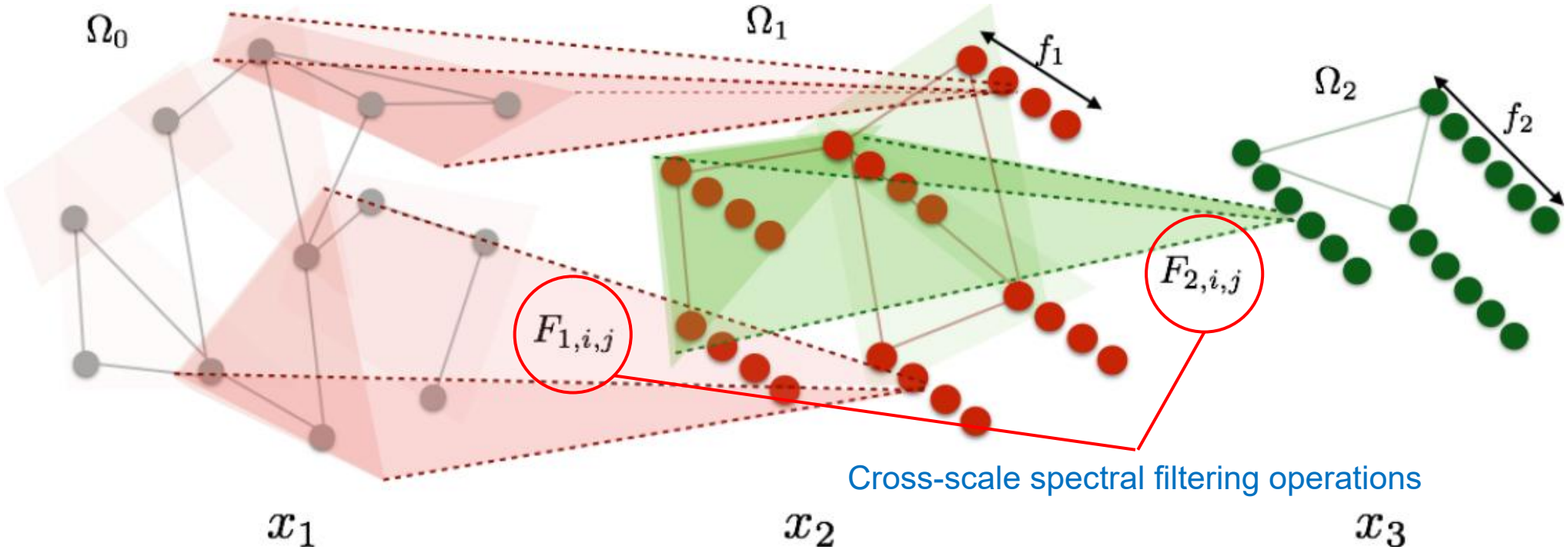
l: node's own features/labels
l_{co}: connected edge features
x_{ne}: neighbouring node states
l_{ne}: neighbouring node labels

[Scarselli et al., 2009]

Node State Computation Through Recursive Neighborhood Aggregation

- Spectral graph CNNs
  - "Spectral Networks and Deep Locally Connected Networks on Graphs" [Bruna et al., 2014]
    - Introduces graph convolutions in spectral domain
    - Establishes connection between graph signal processing and neural networks
    - Proposes learnable spectral filters on graphs
  - Feature handling: Primarily node features
  - Theoretically well-founded in spectral graph theory
  - Require graph Fourier transform through eigendecomposition of the graph Laplacian

Ω_0: Coarse-grained graph scale (lowt resolution scale)
Ω_1: Intermediate graph scale (medium resolution scale)
Ω_2: Fine-grained graph scale (hight resolution scale)



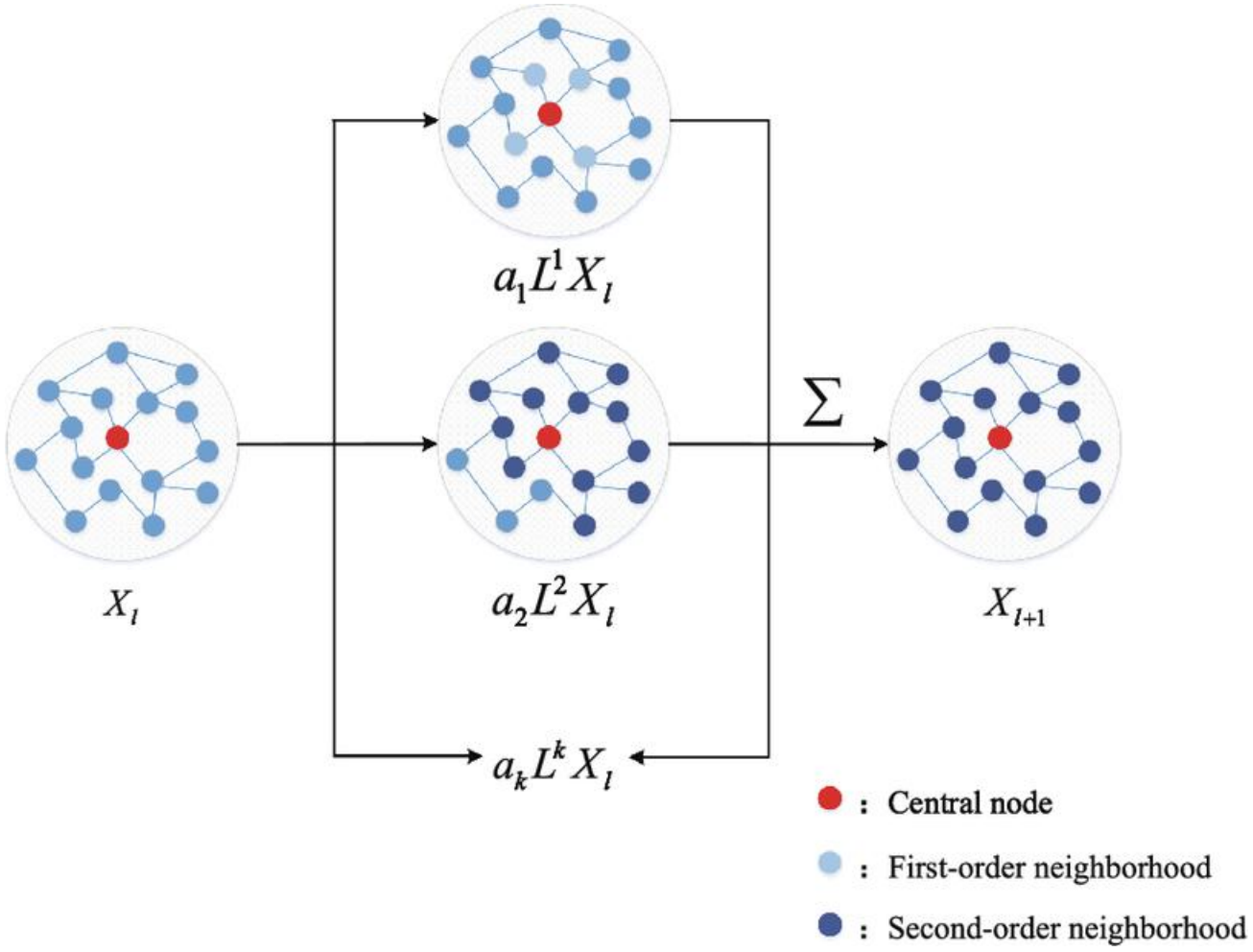
[Bruna et al., 2014]

Spectral Filtering Operations Across Multiple Graph Scales

- ChebNet
  - "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering" [Defferrard et al., 2016]
    - Introduces Chebyshev polynomials for spectral graph convolutions
    - Proposes localized filtering in spectral domain
    - Provides theoretical foundation for later GCN development
  - Feature handling: Node features with K-order spectral filters
  - Focuses on spectral representation of node features
  - Uses K-th order (typically K = 5~25) Chebyshev polynomials (how far (up to K hops) a node can gather information from its neighbors in the graph) to avoid explicit computation of eigenvectors by approximating graph spectral filters

$a_1L^1X_t$, $a_2L^2X_t$, ⋯:

- Suggests polynomial approximation of Laplacian
- Matches the basic form of various polynomial approximation methods, including Chebyshev polynomials

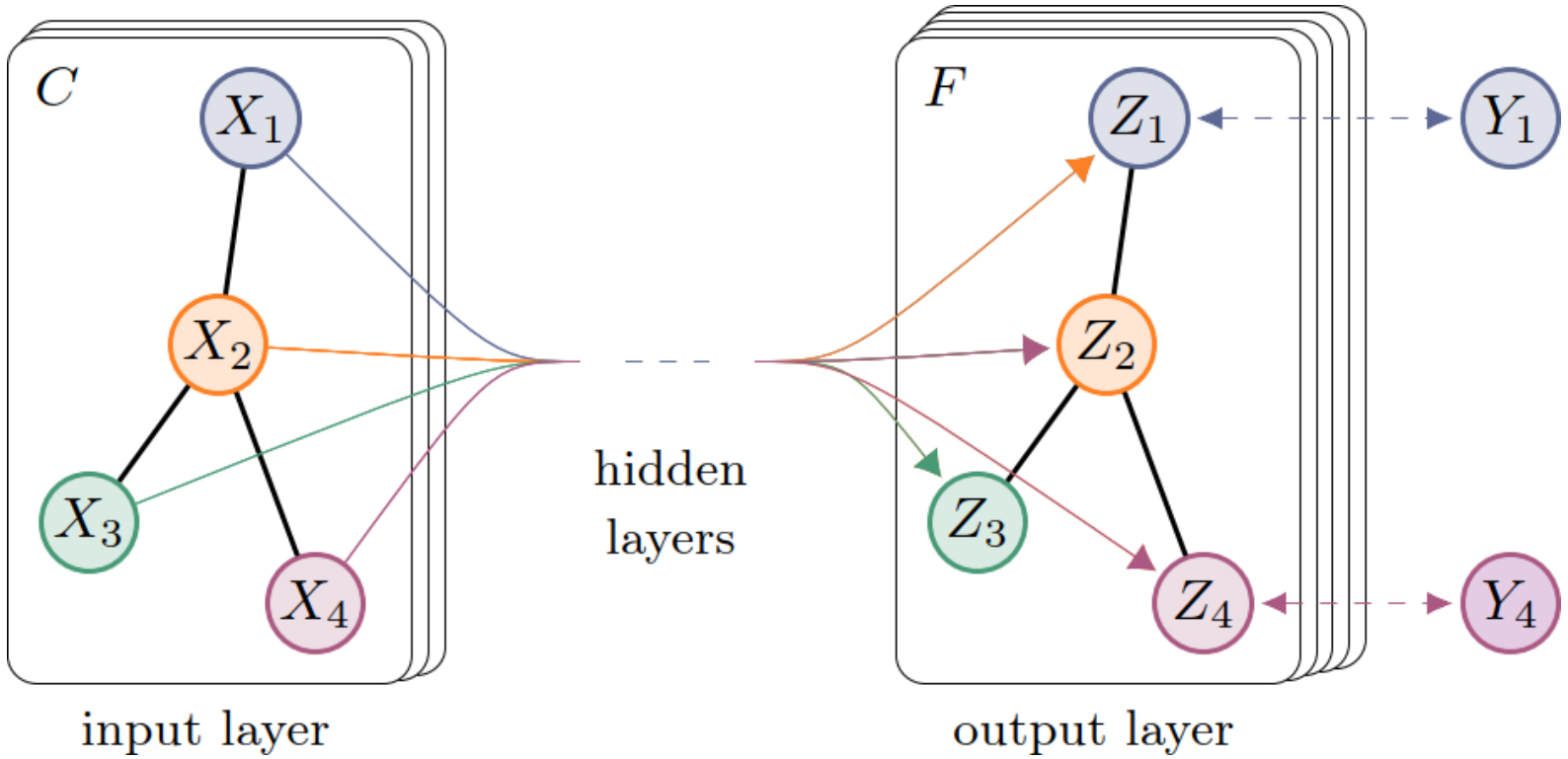


[Xia et al., 2023]

Multi-hop Neighborhood Aggregation Through K-order Graph Convolution

- GCN
  - "Semi-Supervised Classification with Graph Convolutional Networks" [Kipf & Welling, 2017]
    - Proposes localized first-order approximation (further simplification of K=1 (immediate neighbours only) ChebNet) of spectral graph convolutions
    - Establishes efficient layer-wise propagation rule
    - Introduces scalable training for semi-supervised learning
  - Feature handling: Node features with normalized adjacency matrix
  - Computationally more efficient than previous spectral versions
  - Bridges spectral and spatial approaches:
    - Spectral view: Simplified spectral convolution
    - Spatial view: Direct neighborhood aggregation

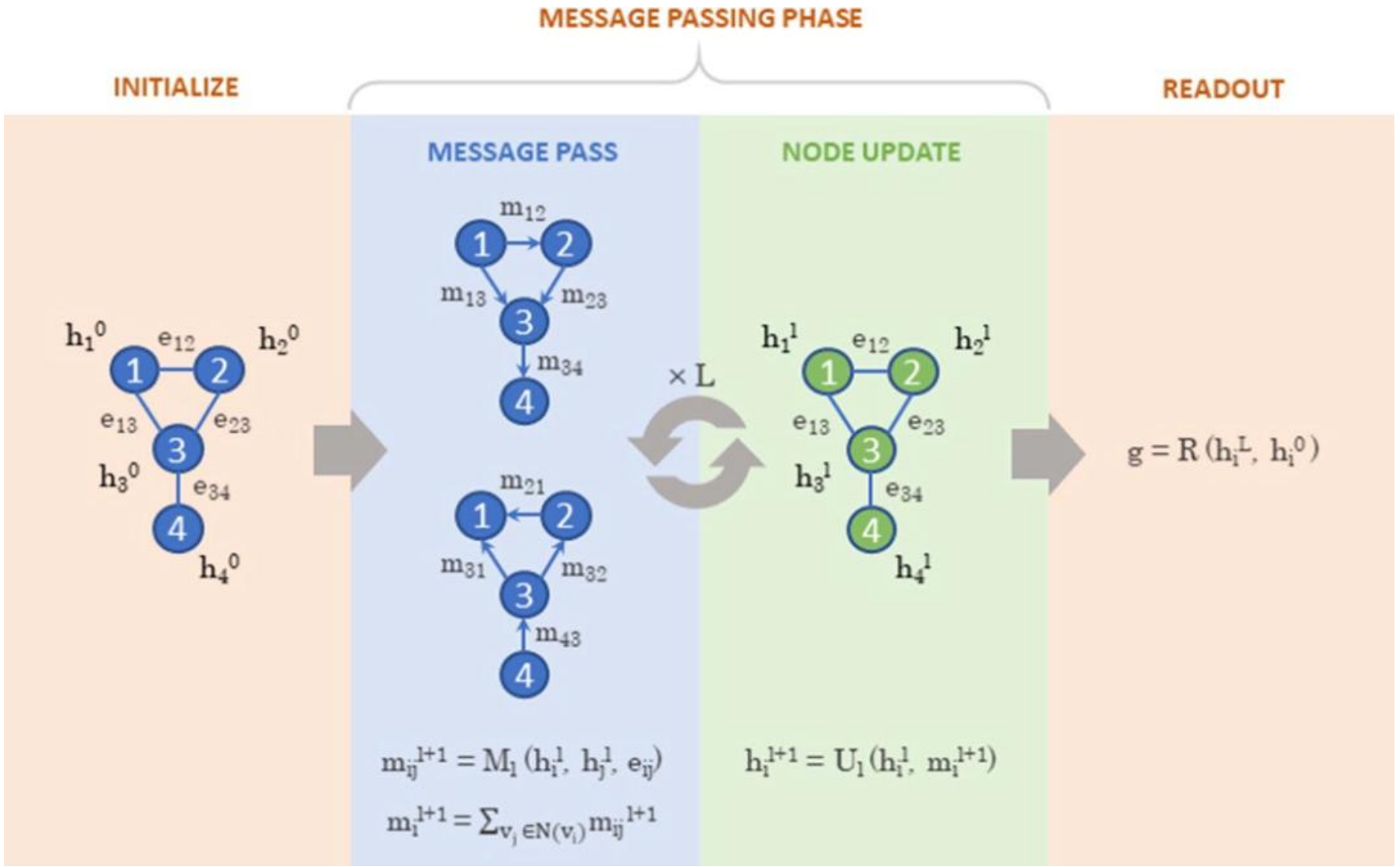
C: Input channels
F: Feature maps



[Kipf & Welling, 2017]

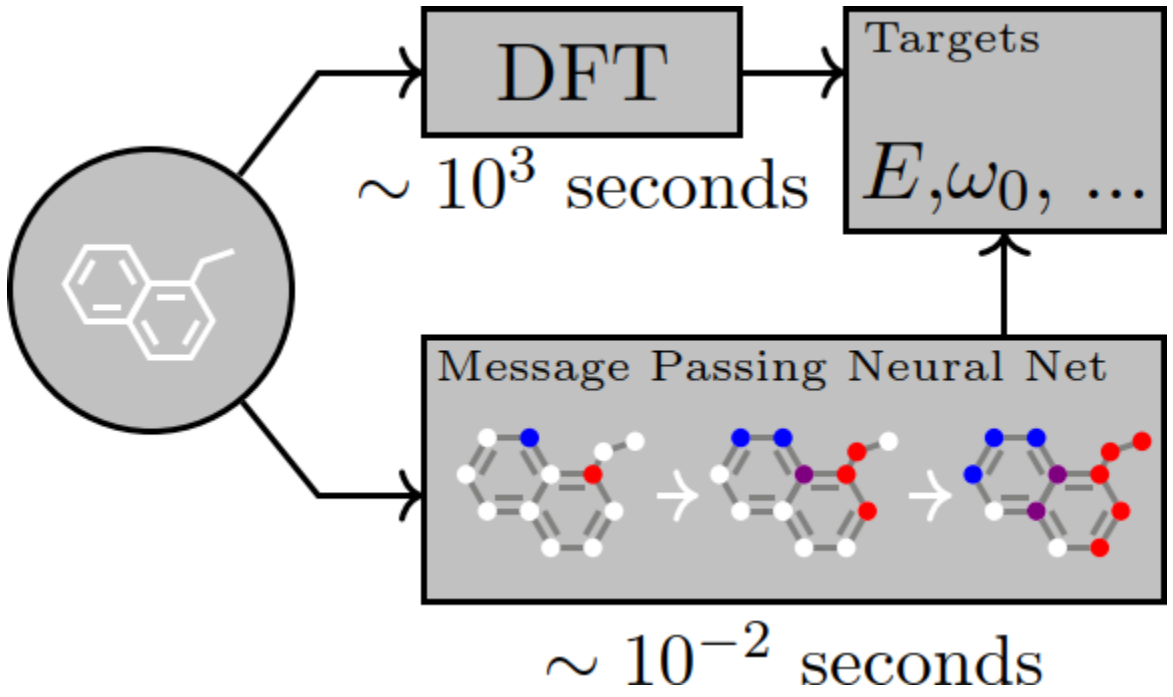
Layer-wise Feature Propagation and Transformation with a Fixed Number of Layers

- MPNN
  - "Neural Message Passing for Quantum Chemistry" [Gilmer et al., 2017]
    - Establishes unified framework for various GNN variants
    - Introduces explicit message passing phases including message, update, and readout (only applicable to graph-level tasks) functions
    - Demonstrates effectiveness in molecular property prediction
  - Feature handling: Both node and edge features naturally
  - Handles different types of nodes and edges through message functions
  - Separates message computation from updates
  - Provides highly flexible framework



[Mercado et al., 2021]

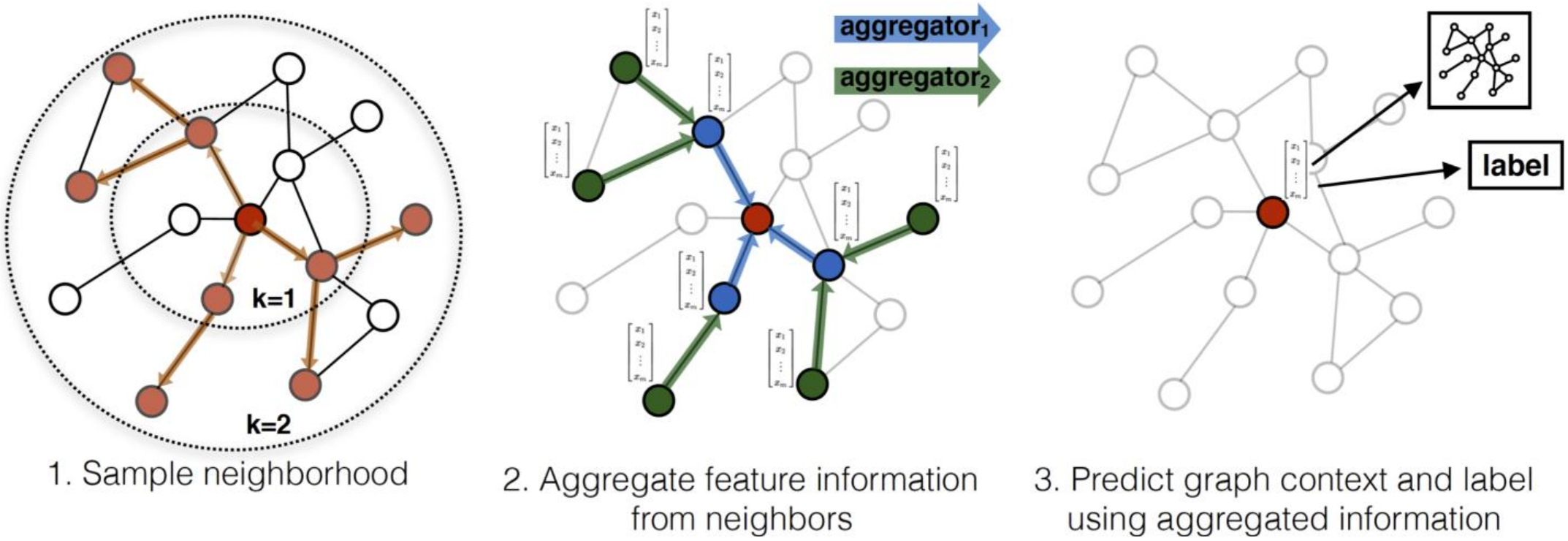
MPNN Schematic



[Gilmer et al., 2017]

MPNN vs. Density Functional Theory (DFT) for Molecular Property Calculations

- GraphSAGE (Graph Sample and Aggregate)
  - "Inductive Representation Learning on Large Graphs" [Hamilton et al., 2017]
    - Introduces sampling-based neighborhood aggregation
    - Proposes multiple aggregation architectures (mean, LSTM, pooling)
    - Demonstrates inductive capability on unseen nodes
  - Feature handling: Node features with sampled neighborhood aggregation
  - Helps shift focus from spectral to spatial approaches
  - Enables inductive learning through local neighborhood sampling and learnable aggregation functions
  - Sets foundation for scalable GNN frameworks



1. Sample neighborhood

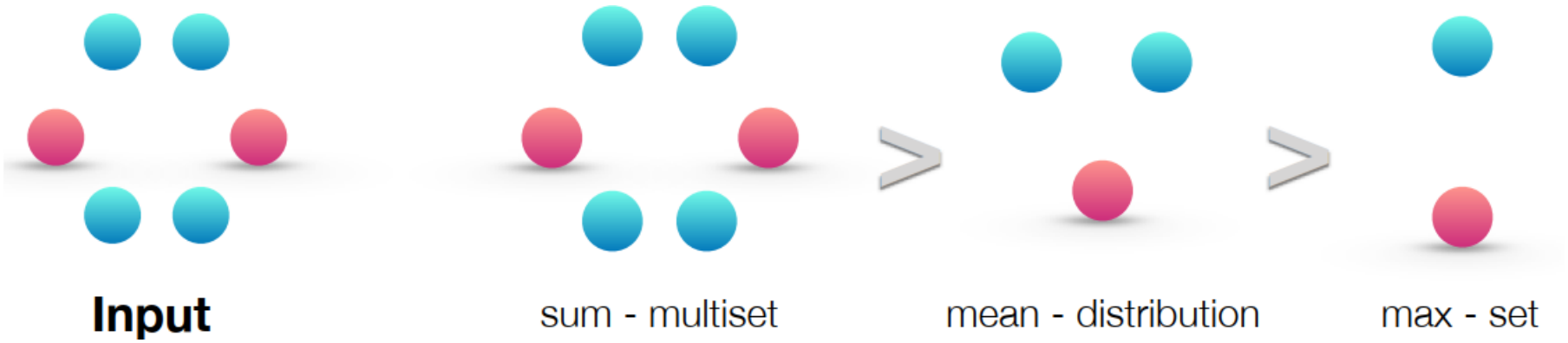
2. Aggregate feature information from neighbors

3. Predict graph context and label using aggregated information

[Hamilton et al., 2017]

Neighborhood Sampling and Feature Aggregation in GraphSAGE

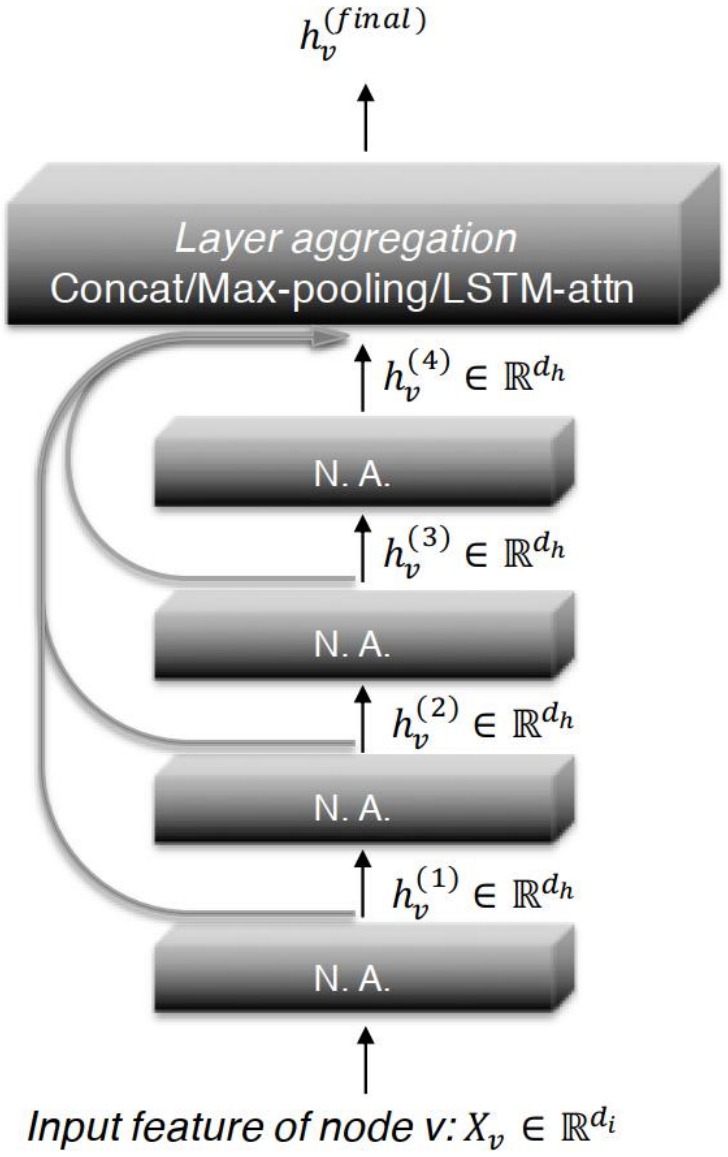
- GIN (Graph Isomorphism Network)
  - "How Powerful are Graph Neural Networks?" [Xu et al., 2019]
    - Proves theoretical expressiveness limits of standard message passing GNNs
    - Shows sum aggregation + multi-layer perceptron (MLP) achieves maximal discriminative power
    - Provides necessary and sufficient conditions for GNN expressiveness
  - Feature handling: Node features with injective (different multisets → different results) aggregation function
  - Most expressive among standard message passing architectures
  - Sum aggregation captures multiset structure of neighborhoods
  - Critical for distinguishing graphs with similar but different structures



[Xu et al., 2019]

Ranking by Expressive Power for Sum, Mean and Max Aggregators over a Multiset

- JK-Net (Jumping Knowledge Networks)
  - "Representation Learning on Graphs with Jumping Knowledge Networks" [Xu et al., 2018]
    - Proposes adaptive layer aggregation: multiple neighborhood ranges per node for better structure-aware representation
    - Offers three aggregation modes: concatenation, max-pooling, LSTM-attention
  - Feature handling: Node features aggregated across multiple layers
  - Enables different nodes to utilize different effective receptive fields
  - Mitigates over-smoothing by preserving representations from all layers
  - Useful for brain networks with heterogeneous connectivity patterns

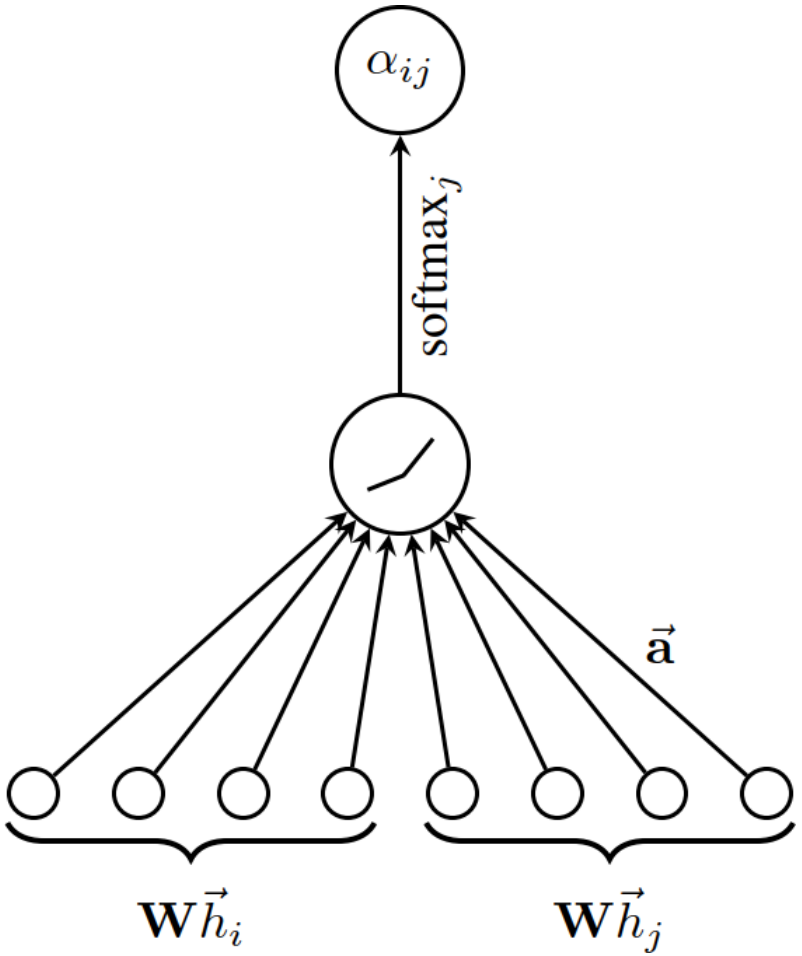


N.A.: neighborhood aggregation

[[Xu et al., 2018]

Aggregation of Multi-hop Neighborhood Information in JK-Net

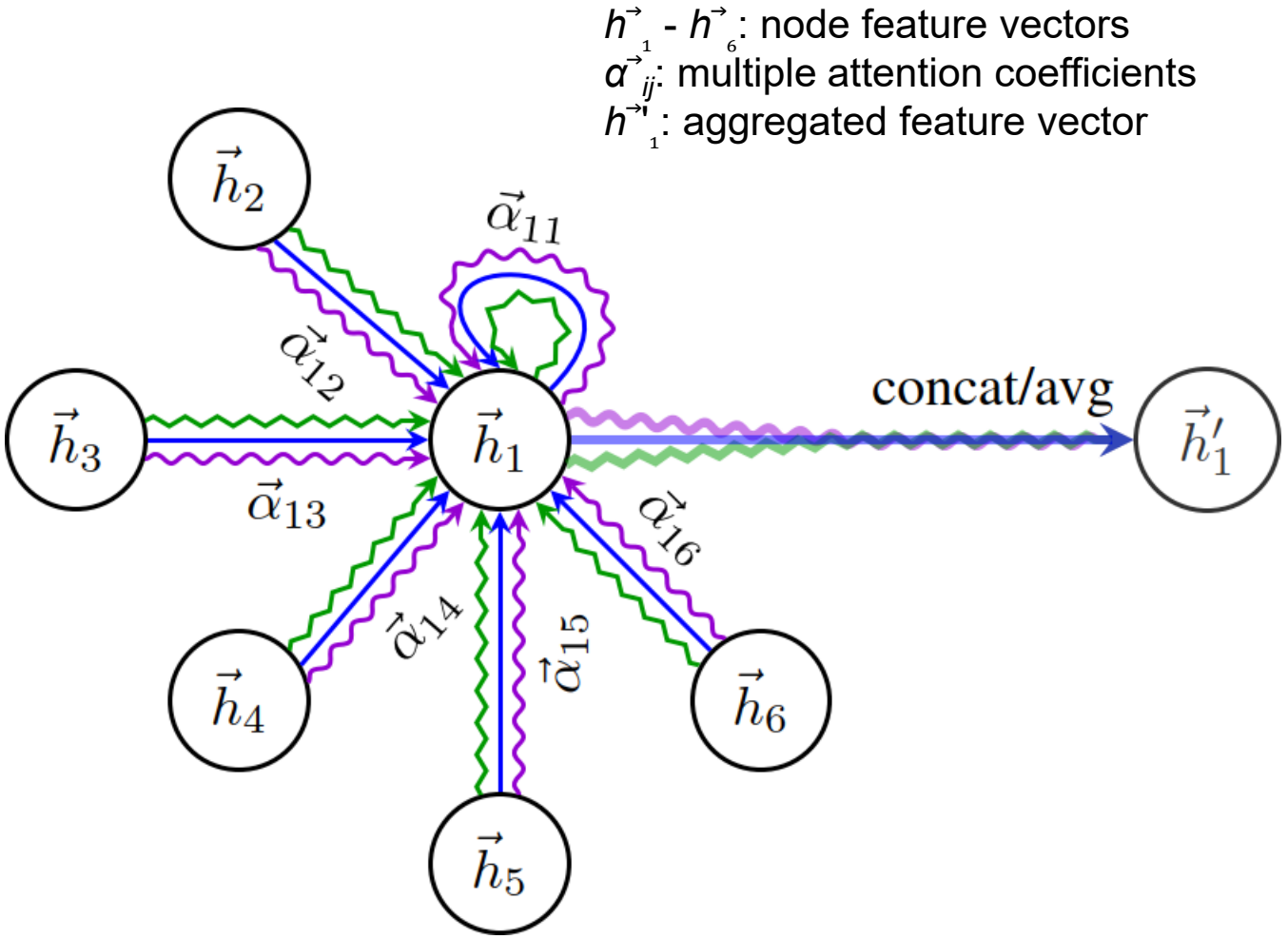
- GAT (Graph Attention Network)
  - "Graph Attention Networks" [Veličković et al., 2018]
    - Introduces attention mechanisms to GNN
    - Proposes learnable neighborhood aggregation weights
    - Establishes multi-head attention for graphs, where multiple independent attention mechanisms operate in parallel, each learning different aspects of node relationships through separate parameterizations and attention coefficients
  - Feature handling: Node features with learned edge importance
  - Better at handling node heterogeneity through attention
  - Generally scalable with attention and adaptive to different graph structures



α_{ij}: attention coefficient between nodes i and j
$\mathbf{W}\vec{h}_i$, $\mathbf{W}\vec{h}_j$: transformed node features
$\vec{a}$: attention mechanism parameter vector

[Veličković et al., 2018]

Attention Coefficient Calculation in GAT



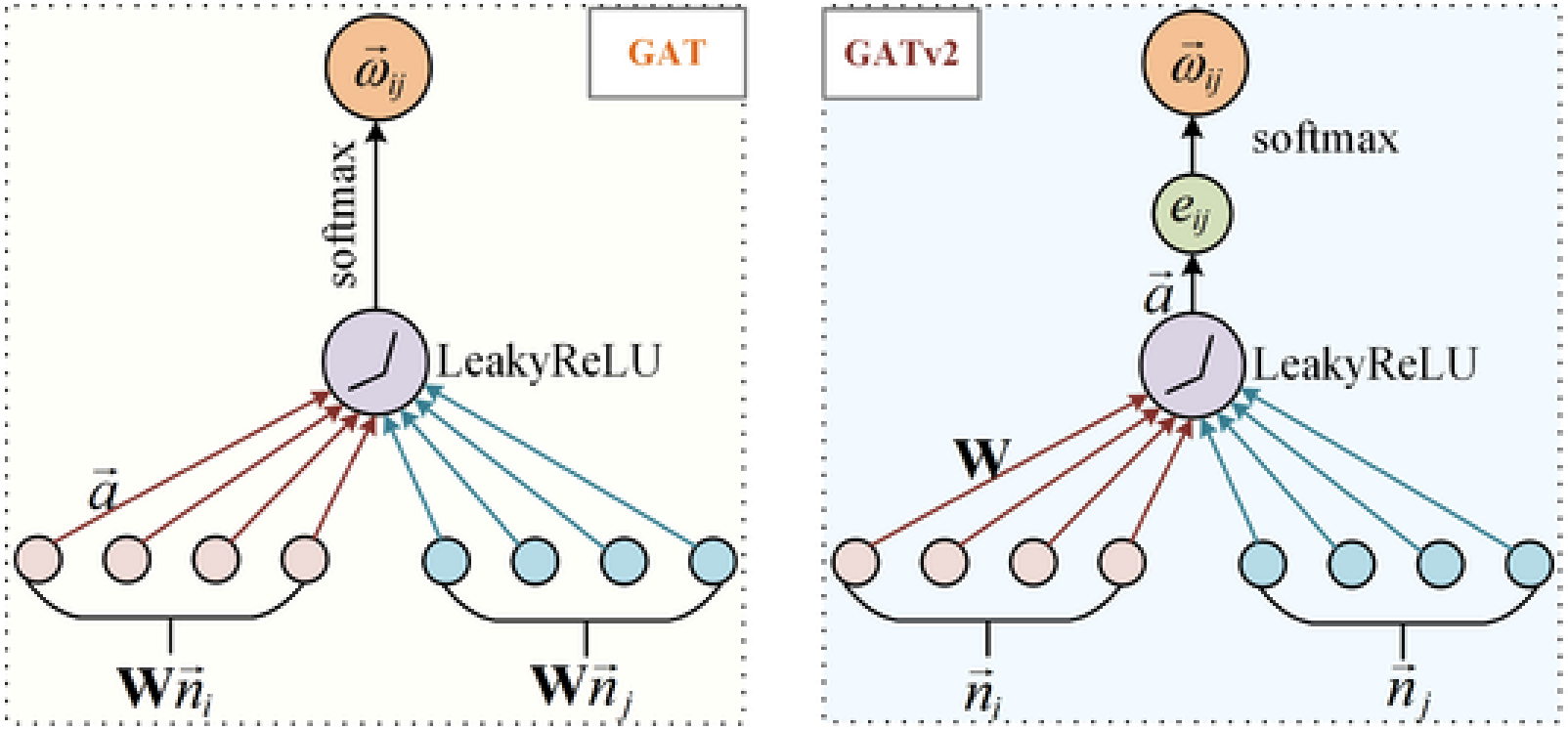
[Veličković et al., 2018]

Multi-head (K = 3) Self-Attention for Node Feature Update

- GAT variants
  - TransformerConv: "Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification" [Shi et al., 2020]
    - Part of UniMP (Unified Message Passaging Model) framework combining GNN with label propagation
    - Implements transformer-style attention within message passing framework
    - Explicitly integrates edge features in attention computation
  - GATv2: "How Attentive are Graph Attention Networks?" [Brody et al., 2022]
    - Fixes original GAT's theoretical limitation of static attention and proposes modified attention mechanism enabling dynamic attention computation
    - Achieves truly dynamic pairwise attention between node pairs

Concatenate → Inner product → Activation

Add → Activation → Inner product



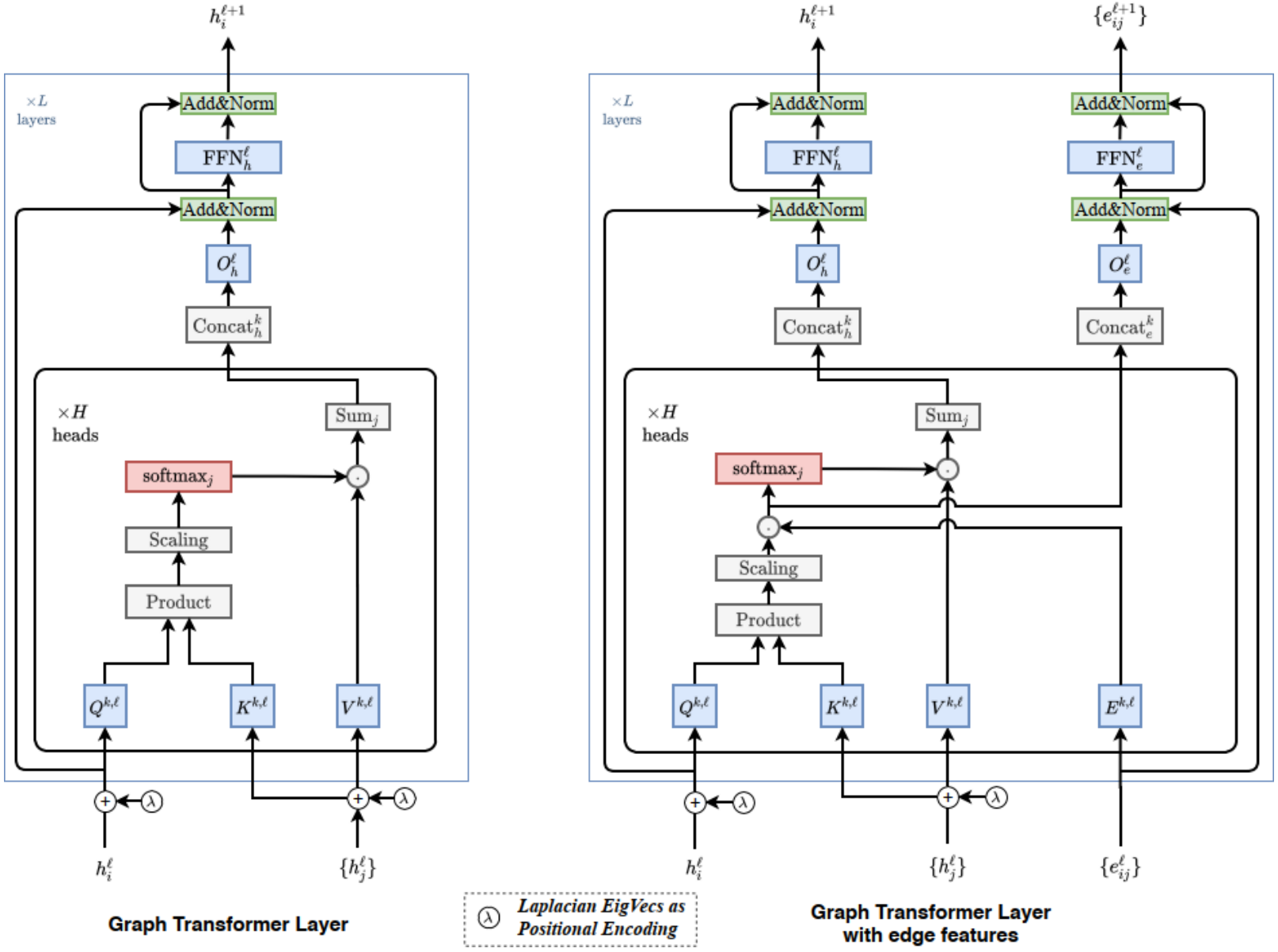
[Liu et al., 2023]

Attention Mechanism of GAT vs. GATv2

- Graph Transformer
  - "A Generalization of Transformer Networks to Graphs"

    [Dwivedi & Bresson, 2021]

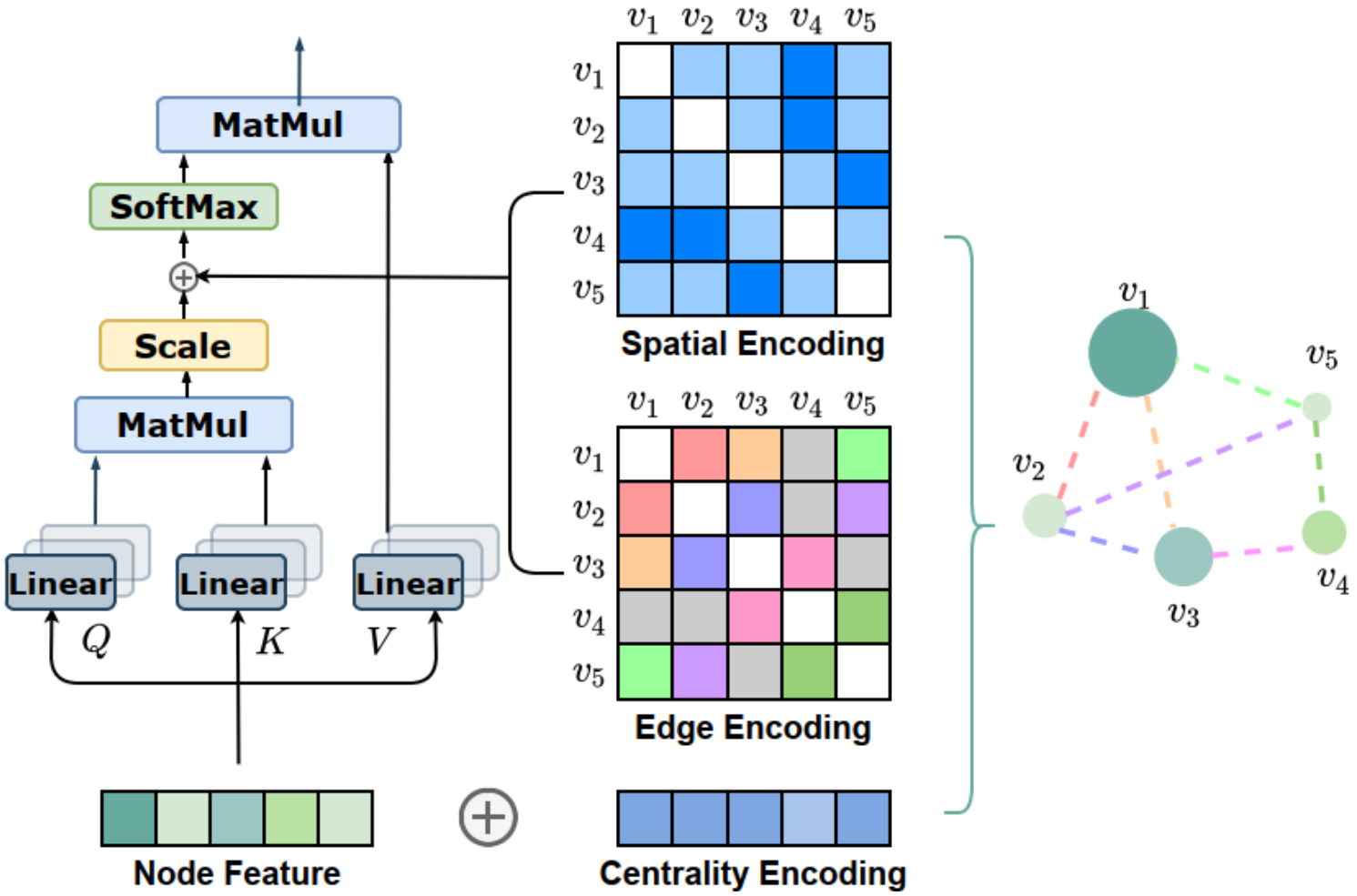
    - Combines local message passing with global attention mechanisms
    - Introduces Laplacian positional encoding for global graph structure
    - Demonstrates viability of transformers for graphs
  - Feature handling: Node and edge features with global attention via Laplacian positional encoding
  - Captures long-range dependencies beyond local neighborhoods
  - Addresses over-squashing problem in deep message passing networks
  - Computational cost, but scalable with sparse attention variants



[Dwivedi & Bresson, 2021]

Graph Transformer Architecture

- Graphormer
  - "Do Transformers Really Perform Bad for Graph Representation?" [Ying et al., 2021]
    - Implements pure transformer approach without message passing layers
    - Introduces graph-specific encodings to standard transformer
    - Shows transformers can excel on graph tasks with proper design
  - Feature handling: Node features with centrality encoding; edge features and graph structure encoded as attention bias via shortest path distances
  - Demonstrates transformers' potential for graphs, inspiring many follow-up works
  - Handles graphs of various sizes and structures



[Ying et al., 2021]

Graph-specific Encodings in Graphormer

- Graph foundation models
  - Self-supervised pre-training for graphs
    - Masked graph autoencoding for transferable graph representations: "GraphMAE: Self-Supervised Masked Graph Autoencoders" [Hou et al., 2022]
    - Contrastive learning without data augmentation: "SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation" [Xia et al., 2022]
  - Unified foundation models
    - Single model generalizing across diverse graph tasks: "One For All: Towards Training One Graph Model For All Classification Tasks" [Sun et al., 2024]
    - Instruction-tuned large language models for graph reasoning: "GraphGPT: Graph Instruction Tuning for Large Language Models" [Tang et al., 2023]

- Key paradigm shifts
  - From task-specific to task-agnostic pre-training
  - From supervised to self-supervised learning
  - From single-domain to multi-domain generalization
  - Integration with large language models for graph reasoning

Implementation of GNNs

- Common frameworks and libraries [Zhou et al., 2020]

Platform	Link	Reference
PyTorch Geometric	https://github.com/rusty1s/pytorch_geometric	Fey and Lenssen (2019)
Deep Graph Library	https://github.com/dmlc/dgl	Wang et al. (2019b)
AliGraph	https://github.com/alibaba/aligraph	Zhu et al. (2019a)
GraphVite	https://github.com/DeepGraphLearning/graphvite	Zhu et al. (2019b)
Paddle Graph Learning	https://github.com/PaddlePaddle/PGL	
Euler	https://github.com/alibaba/euler	
Plato	https://github.com/tencent/plato	
CogDL	https://github.com/THUDM/cogdl/	
OpenNE	https://github.com/thunlp/OpenNE/tree/pytorch	

Model Explanations: Sensitivity Analysis

- General methodology for understanding model behavior
  - How does model output change with input variations?
  - What features are most influential?
  - How robust is the model to input changes?
- Based on analyzing model's response to input changes
- Encompasses various specific techniques
  - Methods often combine different aspects of sensitivity analysis

- Gradient-based methods
  - Computes gradients of output with respect to input features
  - Provides back-propagation based feature importance
  - Approaches
    - Basic gradients: saliency maps
    - Integrated gradients: path integral of gradients
    - SmoothGrad: averaging gradients over noisy samples
  - Characteristics
    - Computationally efficient
    - Direct access to model gradients required
    - Can be sensitive to local variations
    - May produce noisy results

- Perturbation-based methods
  - Observes model response to input modifications
  - Provides local feature importance through input modifications
  - Approaches
    - Local interpretable model-agnostic explanations (LIME)
    - Feature value modification
    - Noise injection
  - Characteristics
    - Model-agnostic
    - Computationally expensive
    - Results may vary with perturbation strategy

- Theoretical attribution-based methods
  - Attributes predictions to input features
  - Provides global feature importance through theoretical frameworks
  - Approaches
    - Shapley additive explanations (SHAP)
    - Layer-wise relevance propagation (LRP)
    - Deep learning important features (DeepLIFT)
  - Characteristics
    - Theoretically well-founded
    - Computationally expensive
    - Can handle feature interactions

- Occlusion-based methods
  - Special case of perturbation-based methods focusing on systematic removal or masking
  - Provides feature importance through systematic removal effects
  - Approaches
    - Occlusion sensitivity analysis
    - Meaningful perturbation
    - Extremal perturbation
  - Characteristics
    - Model-agnostic
    - Computationally expensive
    - Results may vary with occlusion strategy

Occlusion Sensitivity Analysis

- Systematically measures the impact of removing or masking specific parts of the input on model predictions, providing insights into feature importance and model behavior
- Basic approaches
  - Zero-masking: setting features to zero
  - Pattern masking: replacing features with patterns/noise

- Domain-specific approaches
  - Image data
    - Patch-based occlusion
    - Semantic segment masking
  - Sequential data
    - Time window masking
    - Feature sequence masking
  - Graph/network data
    - Edge masking
    - Node masking
    - Subgraph/component masking

- Impact measurement
  - Performance metric changes
    - Regression: Mean absolute error (MAE)
    - Classification: Accuracy
  - Probability/prediction changes
    - Regression: Prediction value differences
    - Classification: Class probability differences
    - Feature activation changes
  - Statistical measures
    - Effect size: Cohen's *d*, Hedge's *g*, Glass's Δ
    - Significance tests: Parametric/non-parametric tests

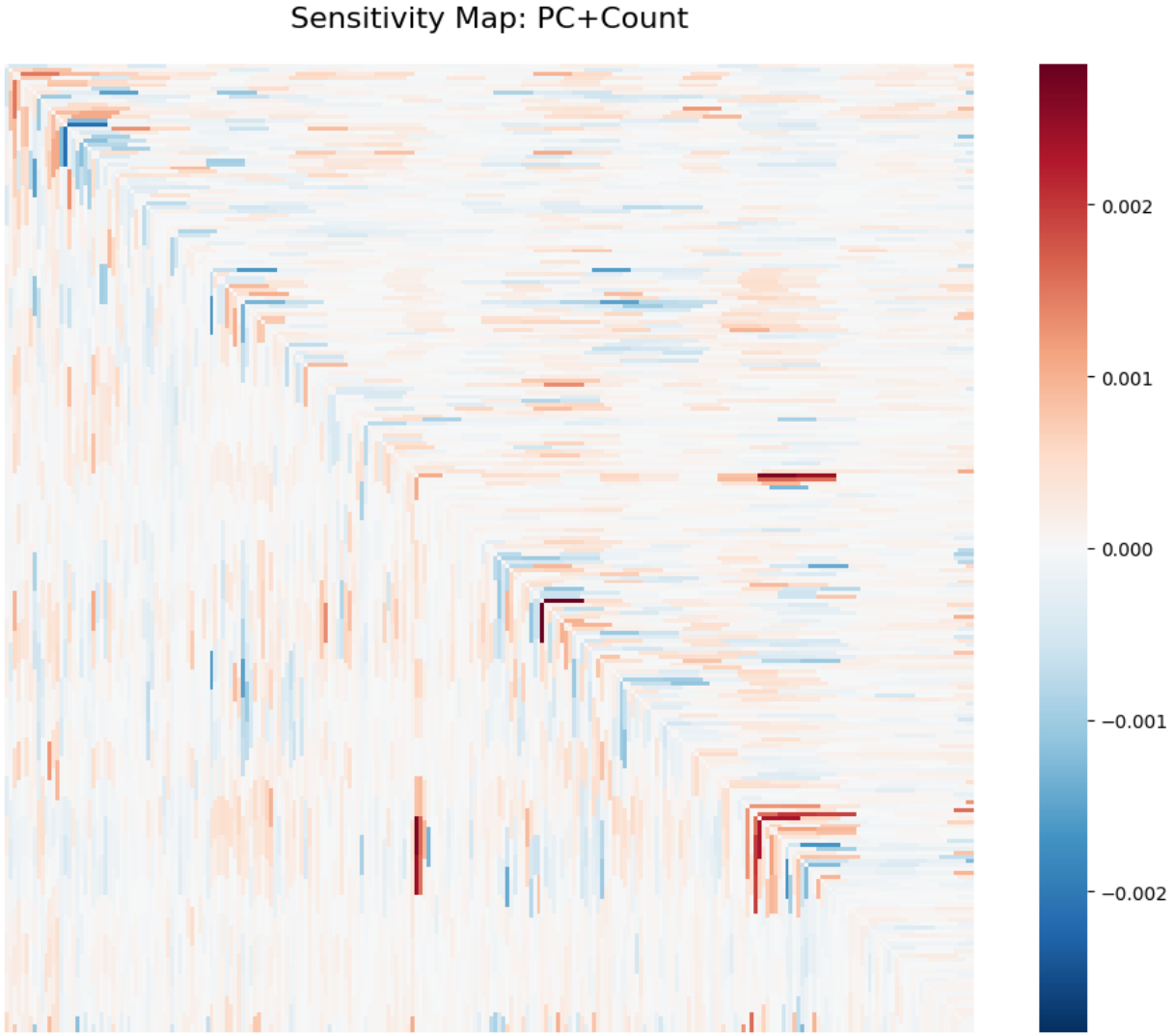
Occlusion Sensitivity Analysis for GNNs

- Graph structure occlusion strategies
  - Edge masking (most common for brain networks)
    - Removes individual edges or set edge weights to zero
    - Reveals critical connectivity patterns
  - Node masking
    - Removes nodes or set node features to zero/mean
    - Identifies important brain regions

  - Subgraph masking
    - Occludes functionally or anatomically defined modules
    - Tests importance of network communities
- GNN-specific considerations
  - Message passing impact: Occlusion affects neighbor aggregation
  - Graph topology changes: Node masking alters network structure
  - Computational cost: O(E) for edge masking, O(N) for node masking
  - Interpretation: Map sensitivity back to brain anatomy

- GNN explainability methods comparison
  - Post-hoc explanation approaches
    - GNNExplainer: "GNNExplainer: Generating Explanations for Graph Neural Networks" [Ying et al., 2019]
      - First general framework for explaining GNN predictions
      - Identifies important subgraph and node features via mutual information maximization
      - Provides both instance-level and model-level explanations
    - PGExplainer: "Parameterized Explainer for Graph Neural Network" [Luo et al., 2020]
      - Learns parameterized explanation network
      - Provides consistent explanations across similar instances
      - More efficient: Single forward pass vs. iterative optimization
    - Occlusion sensitivity analysis: Systematic perturbation-based analysis

- Trade-offs
  - GNNExplainer/PGExplainer:
    - (+) Identifies minimal sufficient subgraphs
    - (−) Requires additional optimization or training
    - (−) May not cover all relevant structures
  - Occlusion sensitivity:
    - (+) Tests all edges/nodes
    - (+) Direct interpretation
    - (−) Computationally expensive (O(E) or O(N) forward passes)
- For brain networks: Occlusion sensitivity analysis preferred
  - Comprehensive: Exhaustive analysis without sampling
  - Interpretable: Direct mapping to brain anatomy
  - Model-agnostic: No additional optimization or training
  - Established: Similar to lesion studies in neuroscience



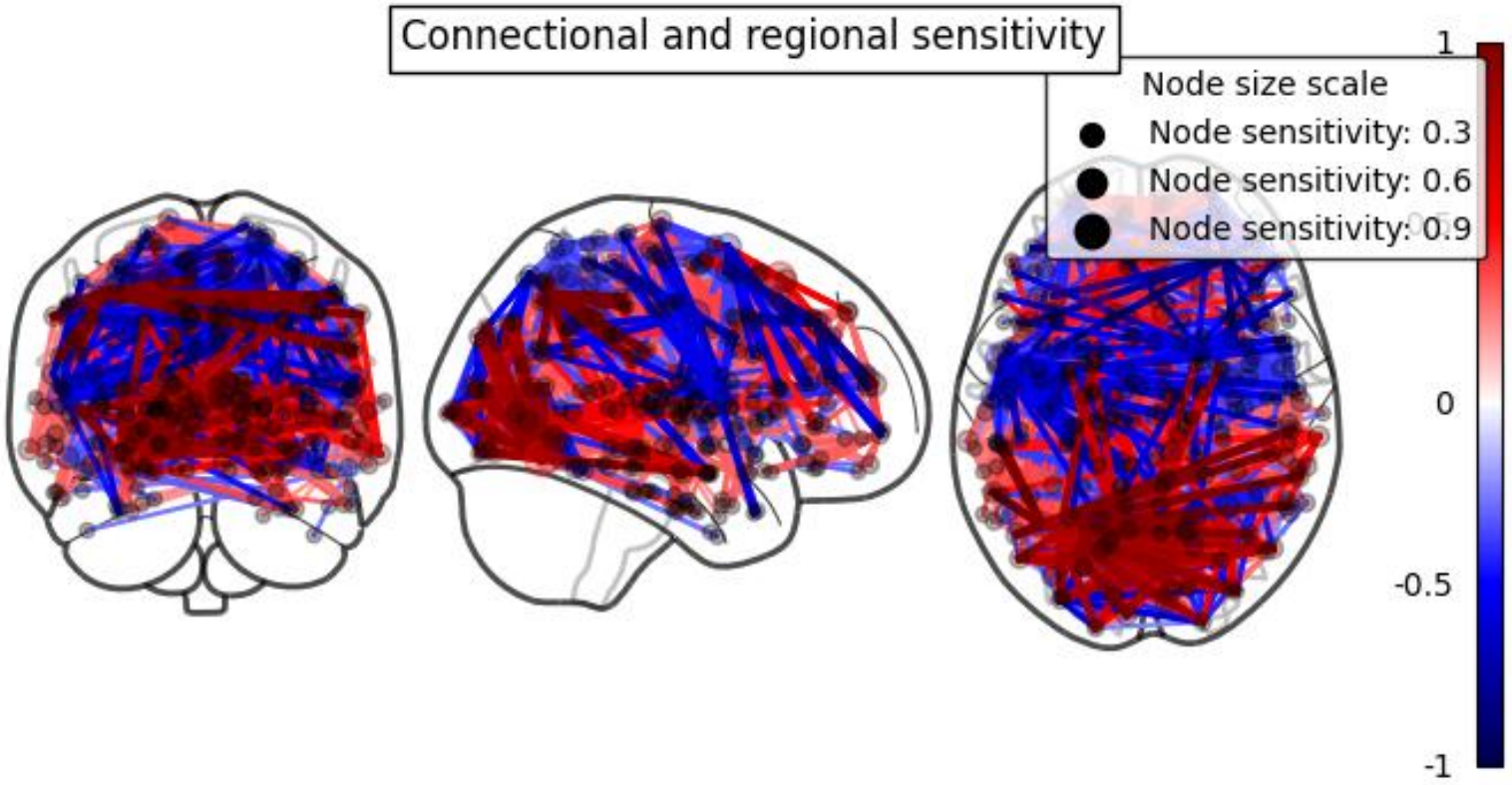
For the calculation,

sensitivity = prediction - baseline prediction:

1. Positive sensitivity (> 0)
- Masking edge → increased prediction
- Edge presence enhances female prediction

2. Negative sensitivity (< 0)
- Masking edge → decreased prediction
- Edge presence enhances male prediction

Sensitivity Map for Brain Networks



Brain Network Representation of Most Sensitive Edges (Top 5% by Magnitude)

Demonstration Experiments

- `torch_geometric.nn`
  - **`GCNConv`** (GCN)
    - Simplified spectral convolution with normalized adjacency matrix for efficient local neighborhood aggregation
    - Sex classification applicability: Effective capture of local brain connectivity patterns through first-order graph filtering
    - Advantages: Computational efficiency with linear complexity, straightforward implementation, robust baseline performance

– **SAGEConv** (GraphSAGE)
  - Sampling-based inductive learning with configurable neighborhood aggregation functions
  - Sex classification applicability: Scalable sex classification across different brain parcellation resolutions and unseen subjects
  - Advantages: Inductive capability for new subjects without retraining, memory-efficient through sampling, cross-dataset generalization
– **GATConv** (GAT)
  - Attention-based neighborhood aggregation with learnable edge importance weights
  - Sex classification applicability: Adaptive weighting of brain connections based on sex-discriminative relevance through attention mechanism
  - Advantages: Interpretable attention weights for connection importance, adaptive to heterogeneous connectivity

- **`TransformerConv`**
  - Transformer-style attention mechanism for local neighborhoods
  - Sex classification applicability: Explicit integration of edge features in attention scoring to capture sex-discriminative connectivity patterns
  - Advantages: Multimodal edge feature integration through attention mechanism, richer representations than GAT while maintaining similar computational cost

- Why use conv layers instead of full models?
  - Conv layers vs. full models in PyTorch Geometric
    - Full models (`GCN`, `GraphSAGE`, `GAT`): Pre-built models with multiple layers
    - Conv layers (`GCNConv`, `SAGEConv`, `GATConv`, `TransformerConv`): Individual building blocks
  - Limitations of full models: Custom components needed for task-specific customization
    - Fixed architecture with hidden implementation details
    - Additional pooling required for graph-level prediction
    - Limited support for multimodal edge features

Conv layer	Edge weight (scalar)	Edge features (vector)	Parameter
GCNConv	○	×	edge_weight
SAGEConv	× (weight=1 for all edges)	×	-
GATConv	○	○	edge_attr with edge_dim
TransformerConv	○	○	edge_attr with edge_dim

Edge Information Utilization in GNNs (PyTorch Geometric)

- Why GNNs take longer to compute despite fewer parameters than CNNs?
  - Irregular nature of graph data structures
  - Sequential aspect of message passing, which cannot leverage the parallelism and optimized memory access patterns

Factor	GNN	CNN
Data structure	Irregular, non-Euclidean graphs with variable neighborhood sizes	Regular grid structures with fixed neighborhood patterns
Information propagation	Sequential node-to-node communication with limited parallelism	Highly parallelizable operations with GPU optimization
Memory access	Sparse, irregular access patterns across graph structures	Dense, contiguous memory access with predictable patterns
Batching	Complex batching of heterogeneous graph structures	Straightforward batching of uniform inputs
Hardware optimization	Limited hardware acceleration for graph operations	Extensive hardware and library support (cuDNN)

GNN vs. CNN: Computational Challenges