STANFORD
SCHOOL OF MEDICINE
Stanford University Medical Center

# *Joint Feature Extraction from Functional Connectivity Graphs with Multi-task Feature Learning*

**André Altmann**, Bernard Ng

Department of Neurology and Neurological Sciences

FIND Lab

altmann@stanford.edu

Stanford, June 11th 2015

# Outline

- Intro
- Multi-task Feature Learning (MFL)
- fMRI Data
- Computational Experiments
  - LOSO-CV performance
  - Feature Stability
  - Edge weights
- Conclusion

# Linear and sparse linear (logistic) regression

- Let $x_i$ be a feature vector of length $p$ and $y_i$ a corresponding label
- When training a linear model, we optimize

$$\min_w \sum_{i=1}^{N} \mathcal{L}(w \cdot x_i, y_i)$$

for the whole dataset with $N$ training instances.

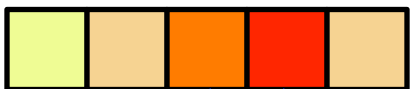
- When $p > N$ we should regularize to avoid overfitting

$$\min_w \sum_{i=1}^{N} \mathcal{L}(w \cdot x_i, y_i) + \lambda \|w\|_q$$

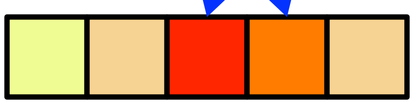
# Interpretation of sparse models (inference)

## Non-sparse ($l_2$-norm)

- Every feature receives a non-zero weight

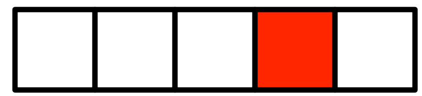


- Data perturbation "redistributes" weights

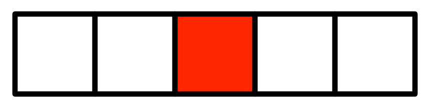


## Sparse ($l_1$-norm)

- Features may receive the weight 0



- Data perturbation "selects" different features



- Use stability selection to get a better idea of frequently selected features

# Sparse learning with multiple tasks

- One task:

$$\min_{w} \sum_{i=1}^{N} \mathcal{L}(w \cdot x_i, y_i) + \lambda \|w\|_1$$

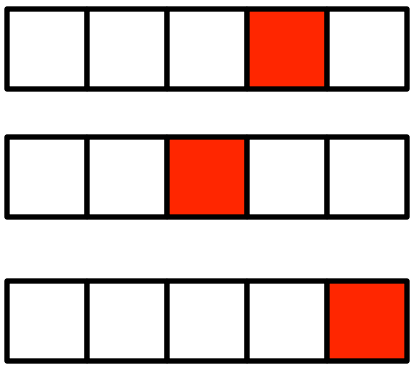
- Naïve extension for *K* tasks:

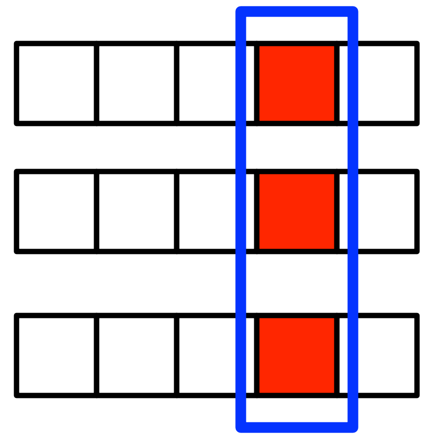
$$\min_{W} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \mathcal{L}(w^k \cdot x_i^k, y_i^k) + \lambda \|w^k\|_1$$

  - We learn *W* instead of *w* across *K* tasks ($w^k$ is *w* for task *k*)

# Sparse learning with multiple tasks

- Each task is still learned independently





$$\|w^k\|_1 \Rightarrow \|W\|_{1,2} = \sum_{j=1}^{p} \|w_j\|_2$$

$l_1/l_2$ - norm

# Multi-task Feature Learning (MFL)

- Instead of the naïve way

$$\min_W \sum_{k=1}^{K} \sum_{i=1}^{N_k} \mathcal{L}(w^k \cdot x_i^k, y_i^k) + \lambda \|w^k\|_1$$

- We optimize

$$\min_W \sum_{k=1}^{K} \sum_{i=1}^{N_k} \mathcal{L}(w^k \cdot x_i^k, y_i^k) + \lambda \|W\|_{1,2}$$

- MALSAR (Zhou, Chen, Ye)

$$\|W\|_{1,2} = \sum_{j=1}^{p} \|w_j\|_2$$

Argyriou et al. (2008, Machine Learning ) Obozinski et al. (2010, Statistics and Computing)

# Experimental fMRI Data

- Dataset from Shirer et al. (2012) CerCor
- Subject-driven cognitive tasks (each 10 min; TR=2s)
  - Resting-state
  - Episodic memory
  - Music and lyrics
  - Counting

- 24 right-handed subjects (age 18-30)
  - N=96 training instances

# Experimental fMRI Data (processing)

- Preprocessing
  - Motion correction
  - Normalized to MNI
  - Spatial smoothing (6mm FWHM)
  - HP filter at 0.008Hz
  - WM
  - CSF
  - Global signal
  - Heart beat, respiration rate

FSL

Regressed out

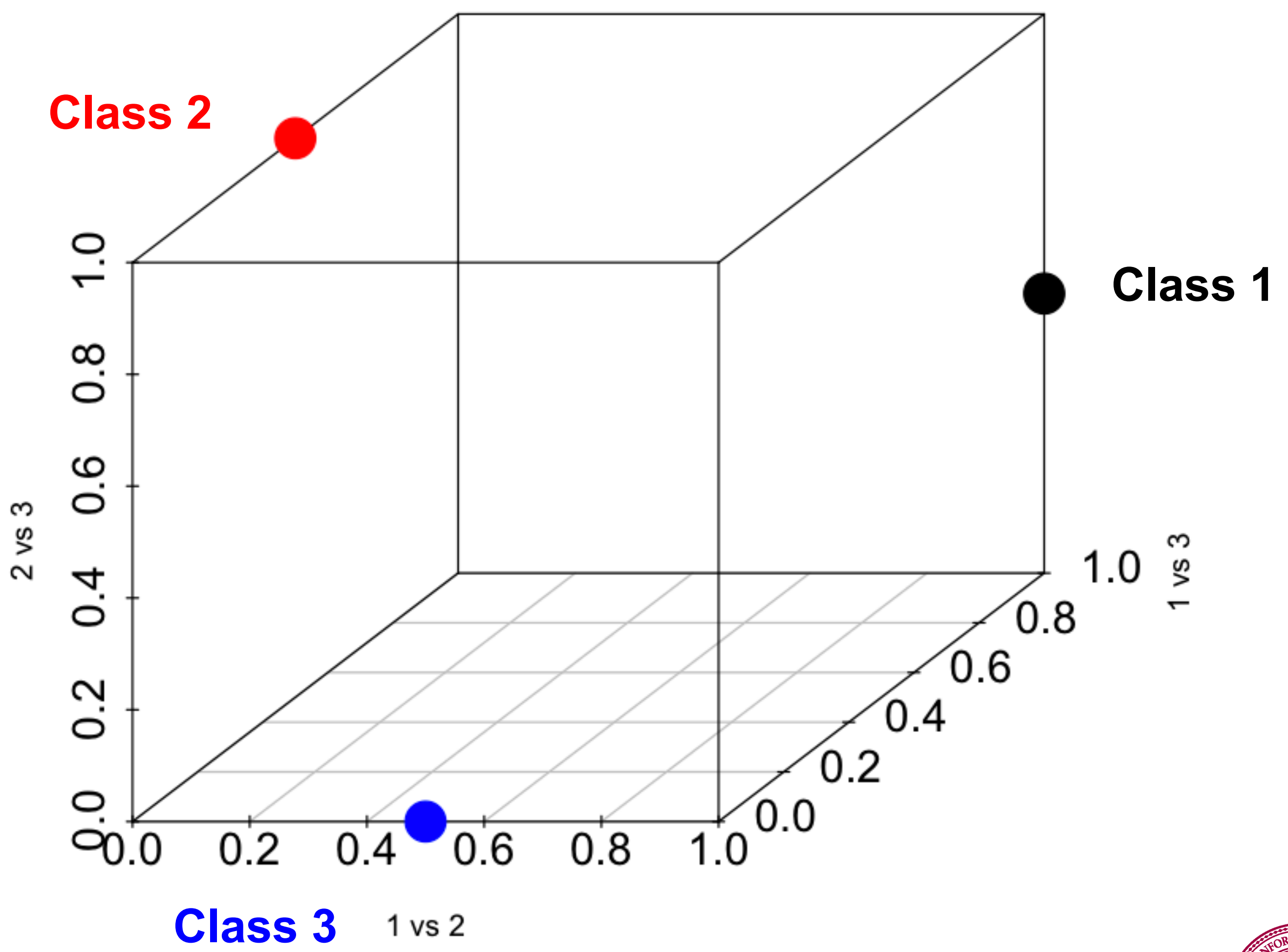
- Feature extraction
  - Time series of 90 ROIs (Shirer et al., 2012)
  - Pairwise Pearson's correlation
  - Converted to z-scores
  - Upper triangle of 90x90 matrix (=4,005 connections)

# Error Correcting Output Codes (ECOC)

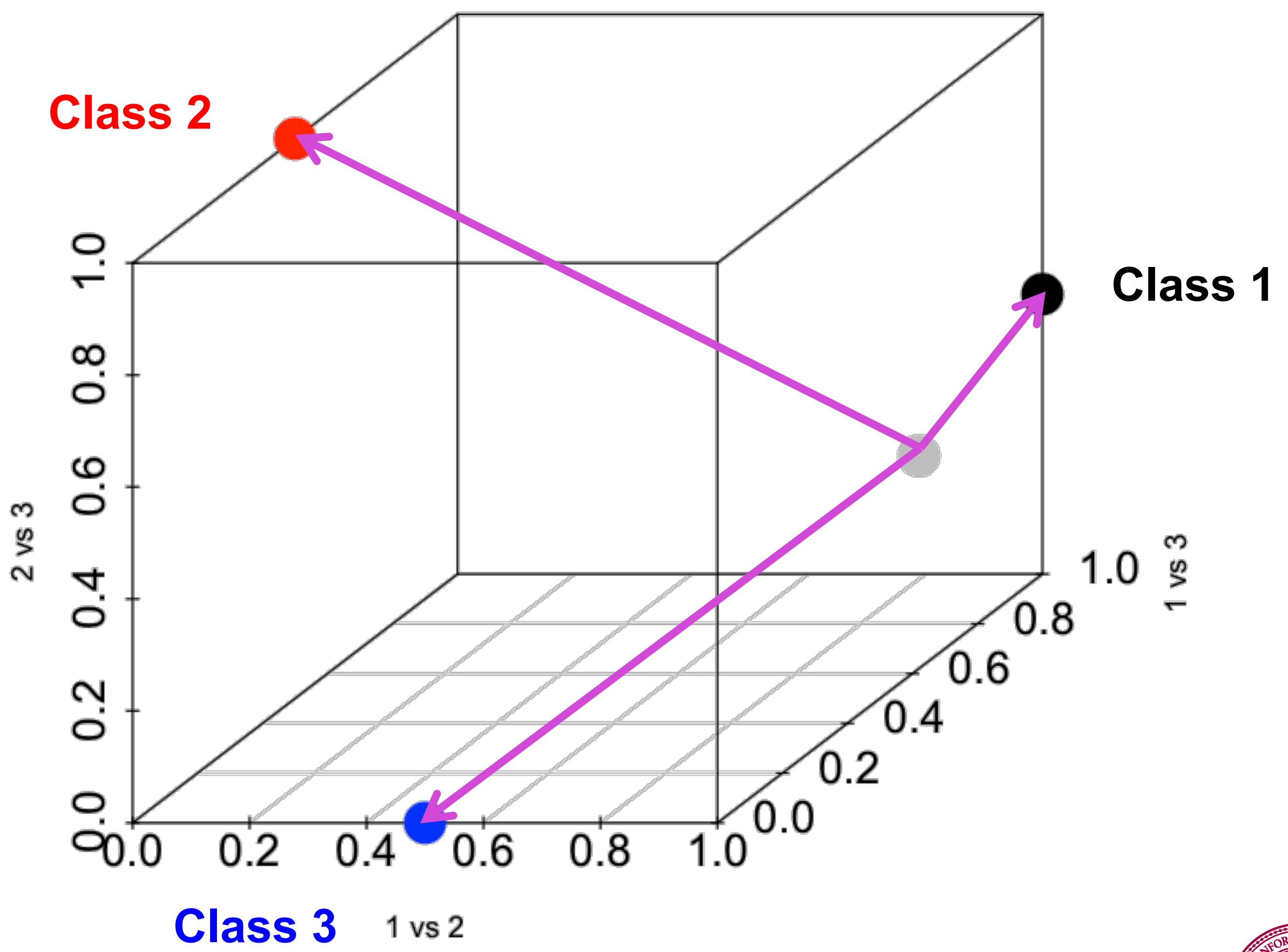
- 4 classes give us 6 different tasks (pair-wise binary)
- We used ECOC to derive a single class from the six outputs
- In a nutshell:
  - Each task provides a prediction [0.0, 1.0]
  - Predictions are compared to "ideal" predictions for one class
    - 3 classes -> 3 tasks (**1 vs 2**, **1 vs 3**, **2 vs 3**)
    - Ideal for class 1: (1.0, 1.0, 0.5)
    - …
  - The "ideal" class vector, which is closest to the observed prediction vector, provides the class label
    - Euclidian distance
    - …

Dietterich and Bakiri (1995)

# Error Correcting Output Codes (ECOC)



# Error Correcting Output Codes (ECOC)



# Experiment 1: LOSO-CV performance

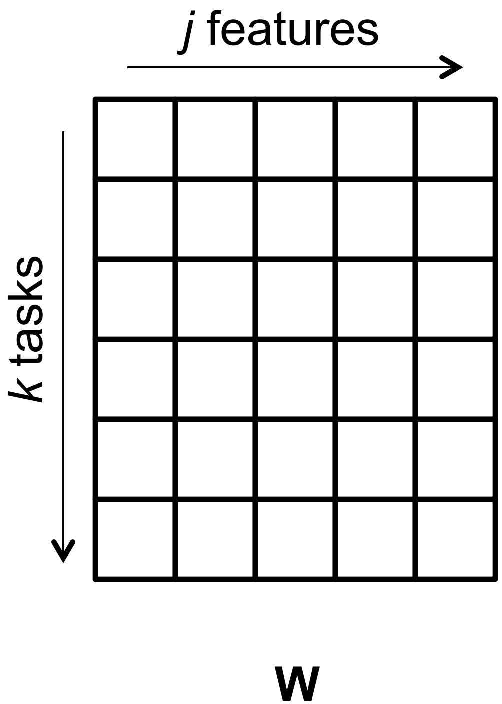
# LOSO-CV classifier performance

- Experimental setup
  - Train 6 classifiers (6 tasks)
  - For each subject:
    - Train on all-but-one subject (LOSO)
    - Predict subject's data
    - Use ECOC to obtain a predicted class label
  - Combine all prediction and compute Cohen's κ
- Screen a range of tuning parameters (λ)
  - $10^{-6}$, $10^{-5.75}$, $10^{-5.5}$, …, $10^{4}$
- Compare to other methods
  - L2 logistic regression (RIDGE): RLR
  - L1 logistic regression (LASSO): SLR
  - Sparse multinomial logistic regression: SMLR

LIBLINEAR

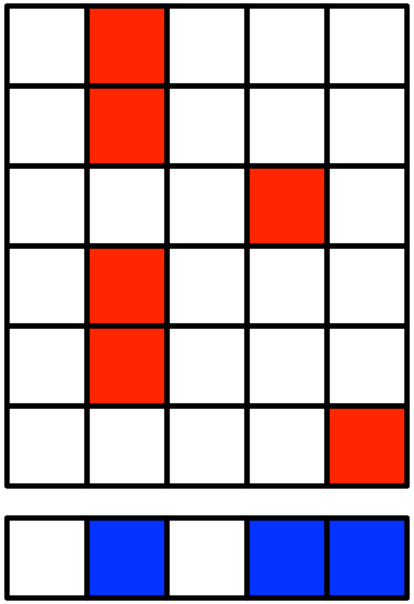
# LOSO-CV classifier performance

- Comparing methods based on $\lambda$ is challenging
- We converted $\lambda$ into number of selected connections

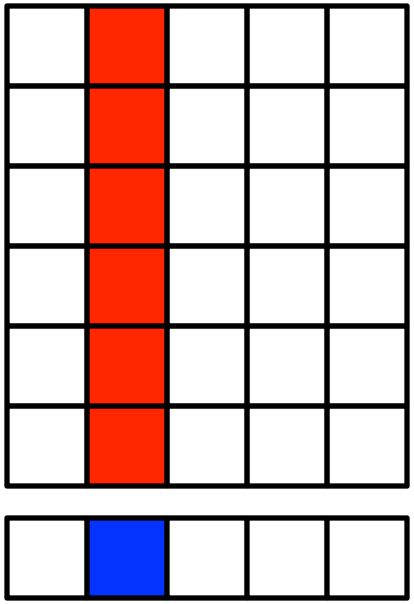


# LOSO-CV classifier performance

- Comparing methods based on $\lambda$ is challenging
- We convert $\lambda$ into number of selected connections



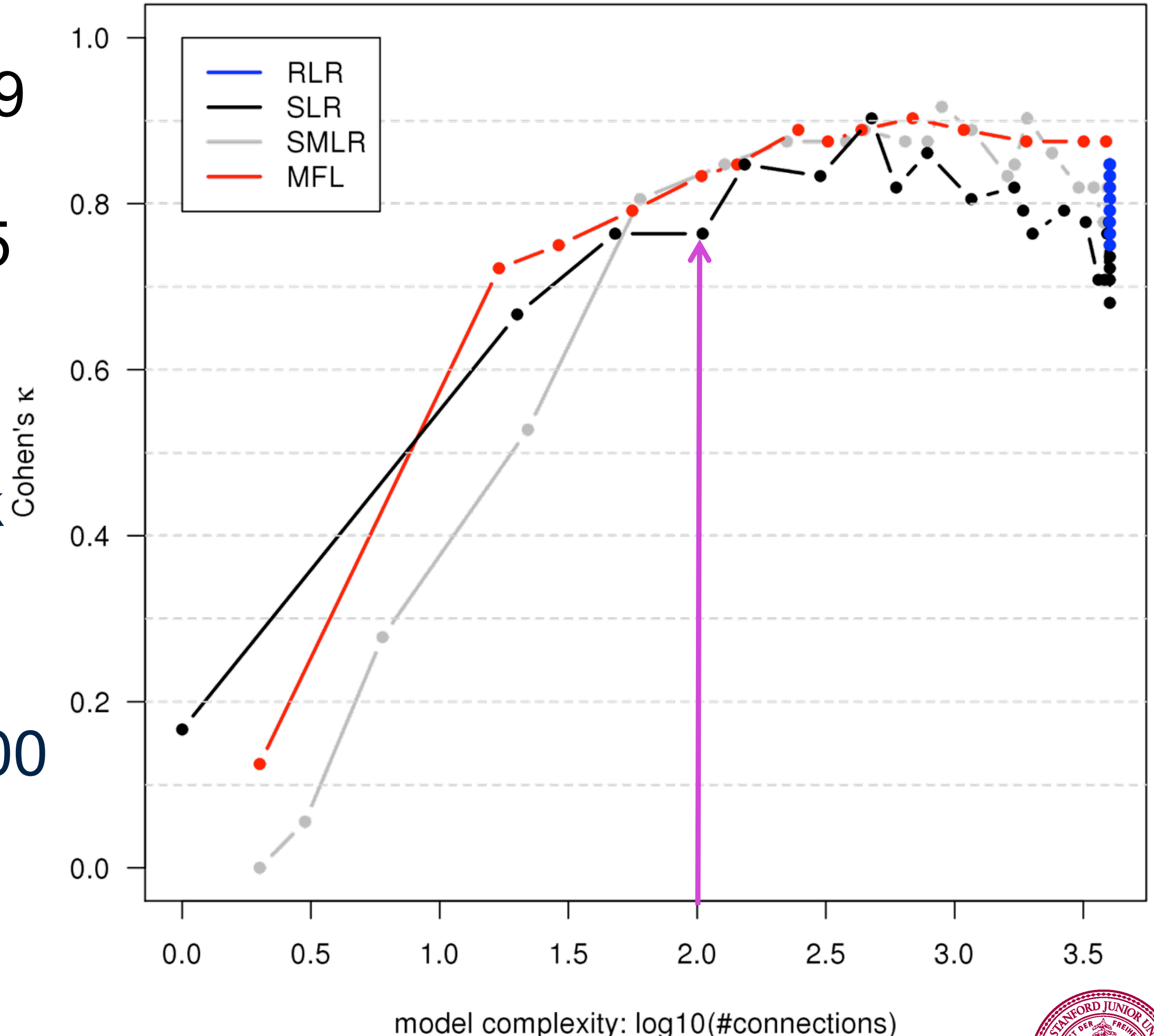
Three connections



One connection

# LOSO-CV classifier performance

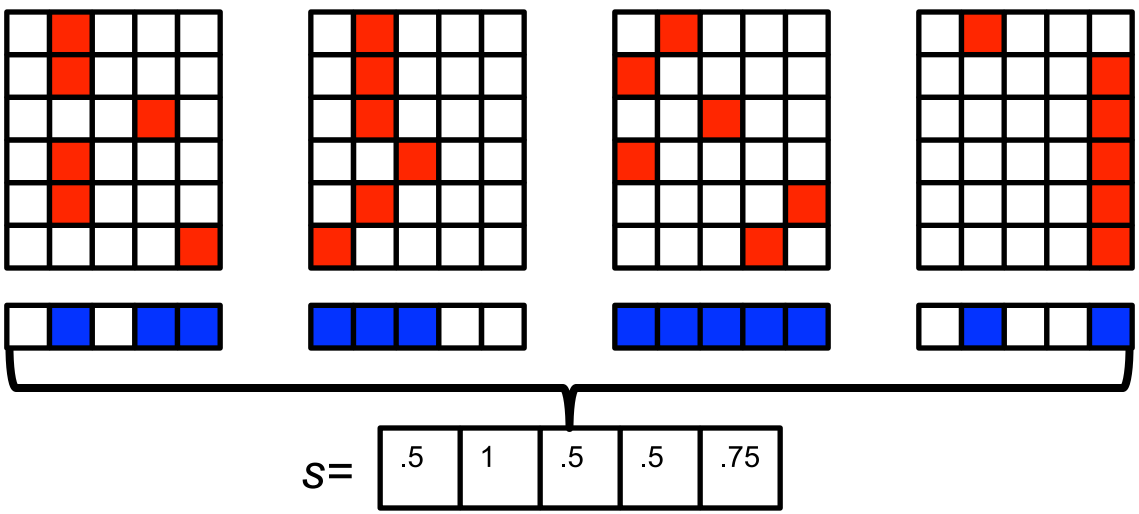
- Max *k* around 0.9
- RLR max at 0.85
- With ~100 conn-ections sparse methods reach k of 0.8
- MFL remains stable with >1,000 connections



# Experiment 2: Feature Stability

# Feature Stability

- Assess reliability of selected edges (features)
  - Perturb training data using bootstrapping with replacement to obtain new training set
  - Train methods on the new dataset using multiple tuning parameters
    - $10^{-6}$, $10^{-5.75}$, $10^{-5.5}$, …, $10^{4}$
  - 100 repetition and compute selection frequency for each feature



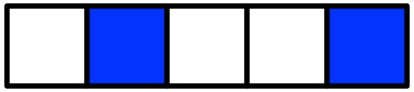
# Feature Stability

- **Assess reliability of selected edges (features)**
  - Perturb training data using bootstrapping with replacement to obtain new training set
  - Train methods on the new dataset using multiple tuning parameters
    - $10^{-6}$, $10^{-5.75}$, $10^{-5.5}$, …, $10^{4}$
  - 100 repetition and compute selection frequency for each feature

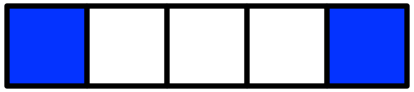
$s=$

| | | | | |
|---|---|---|---|---|
| .5 | 1 | .5 | .5 | .75 |

  - Train classifier on unperturbed data
    - Compute fraction of edges selected >90%



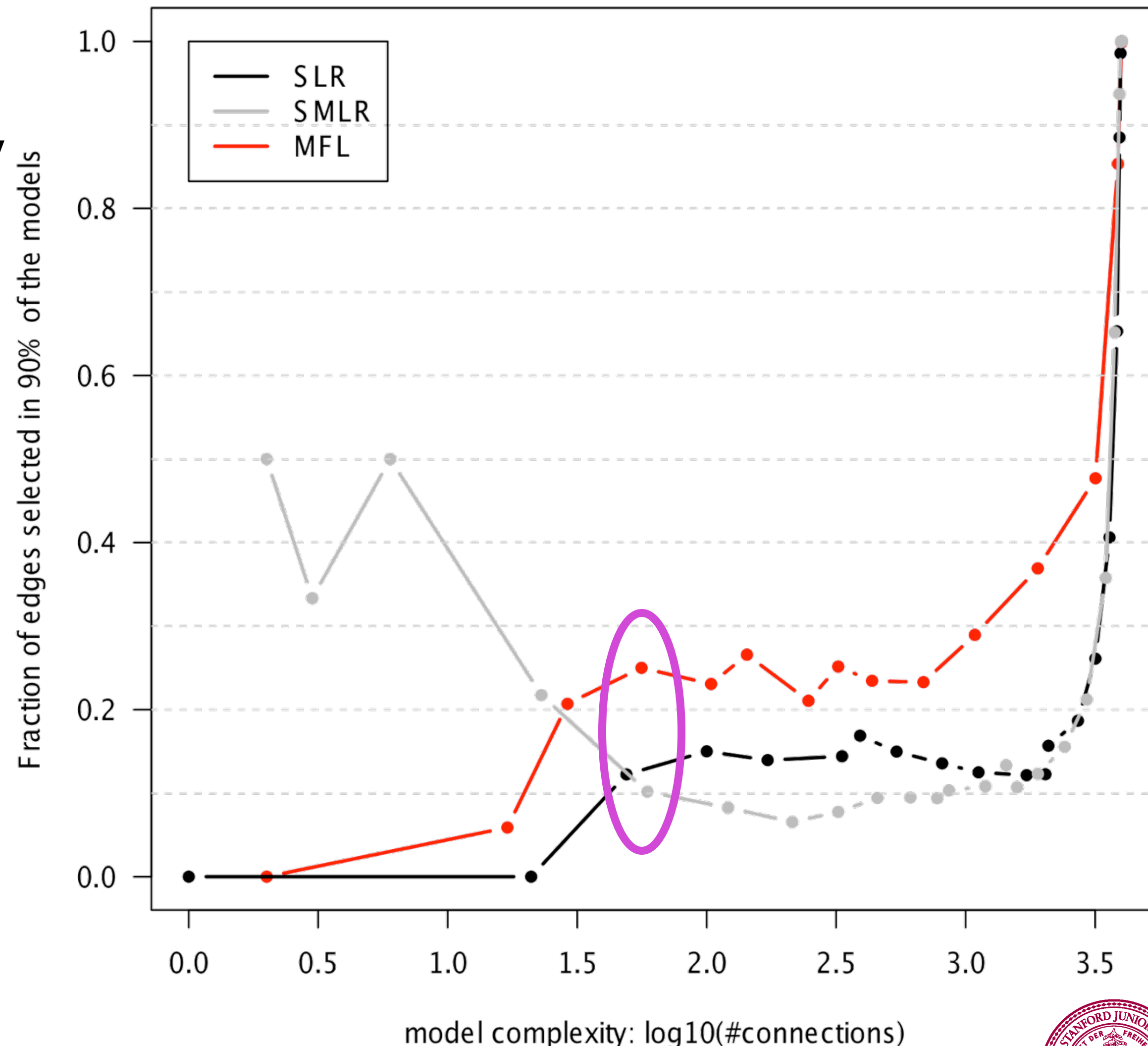
-> 0.5



-> 0.0

# Feature Stability

- SLR and SMLR perform similarly
- MFL better than both in range of 50-2,000 connections
- At 50 edges, 25% for MFL vs ~10% for SLR and SMLR

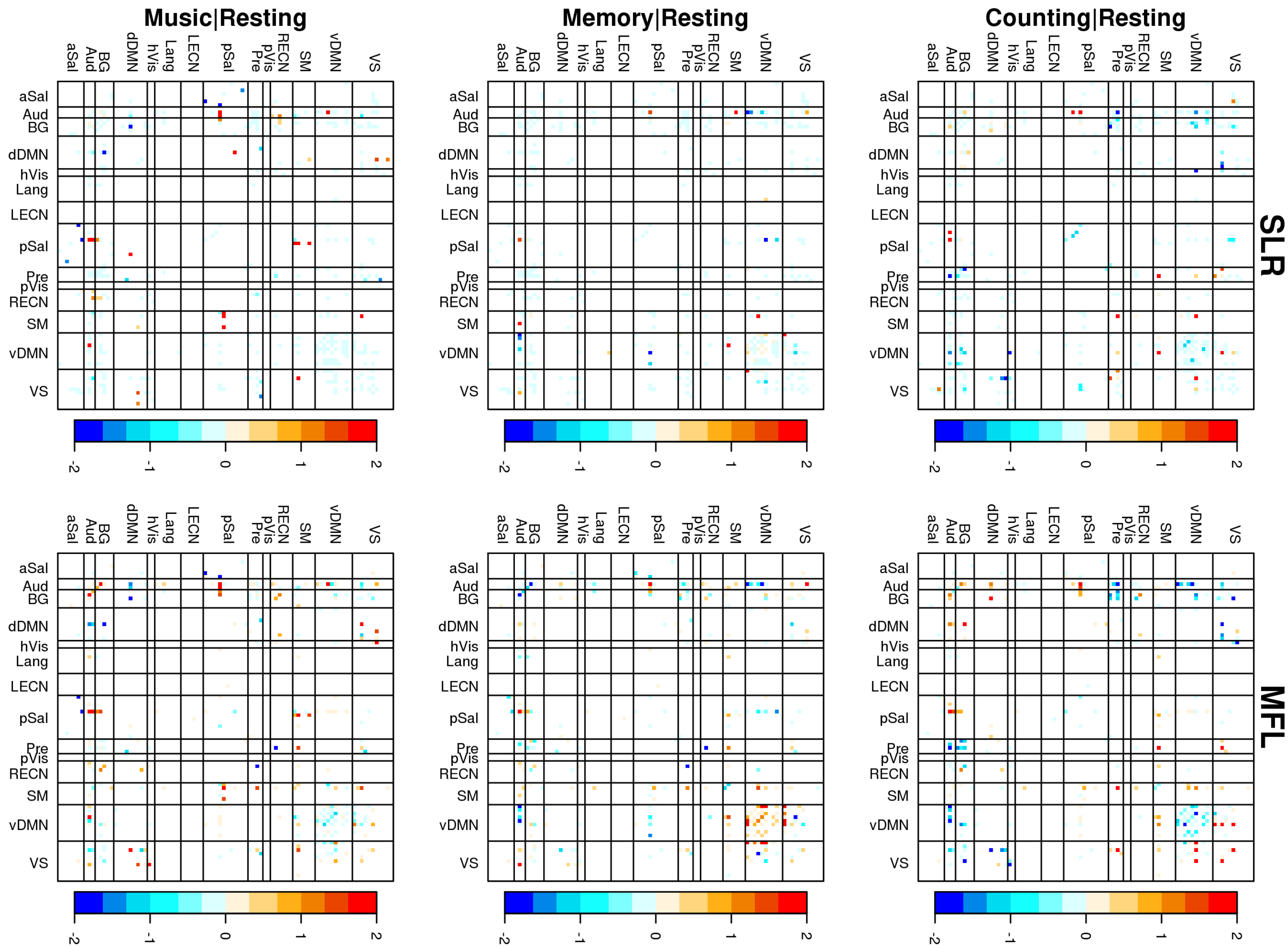


# Experiment 3: Edge weights

# Edge weights

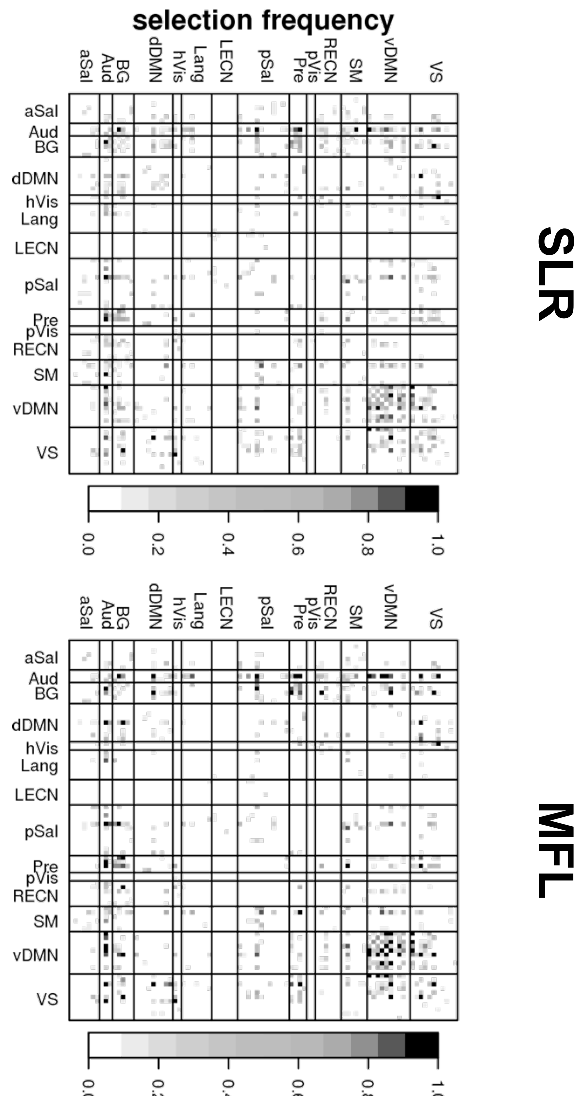
- Visualize weights from MFL and SLR
- Both models have around 150 connections
  - K about 0.84
- For better visualization weights are scaled to mean=0 and sd=1

# Edge weights



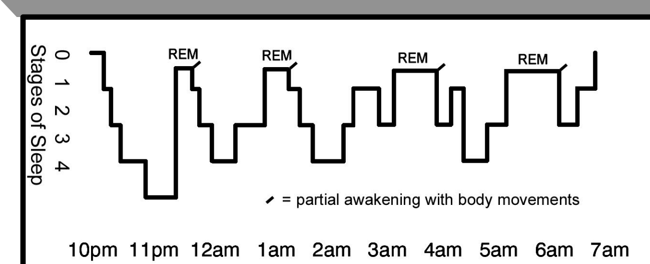
# Edge weights

- Selection frequency higher in MFL than in SLR
- P=0.0017 (Wilcoxon signed-rank test)

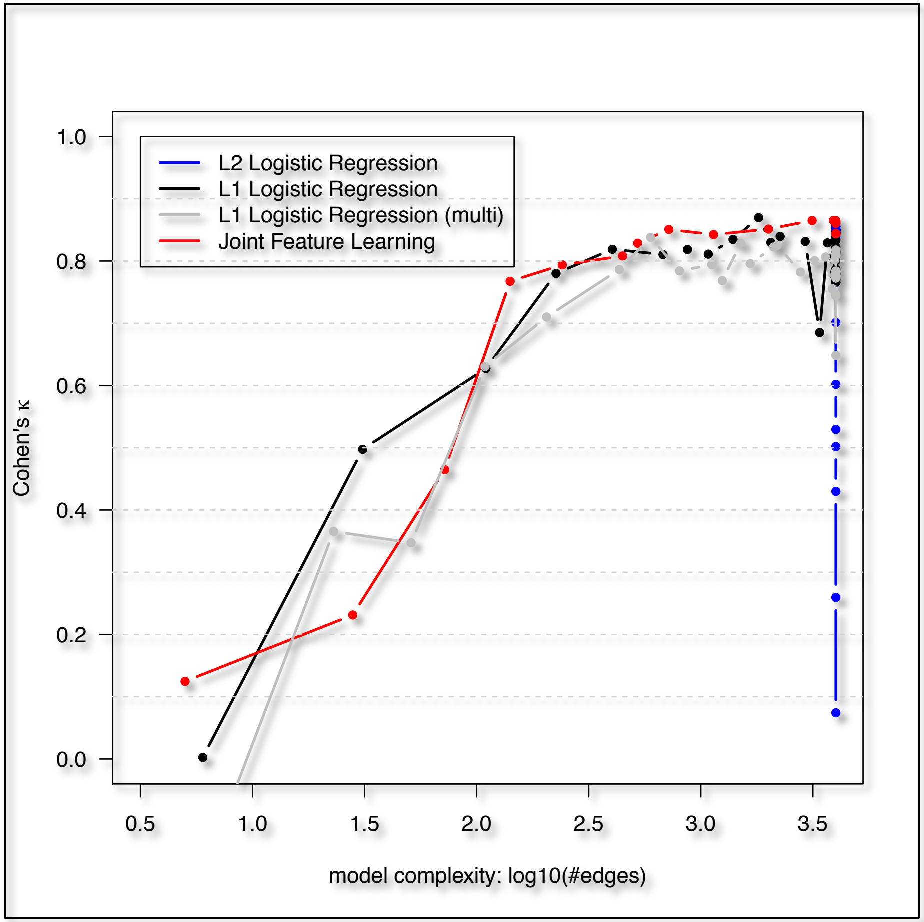


# Predict wakefulness and sleep from rs-fMRI

- Awake (S0)
- Asleep
  - S1 (stage 1)
  - S2 (stage 2)
  - SW (stage 3&4)

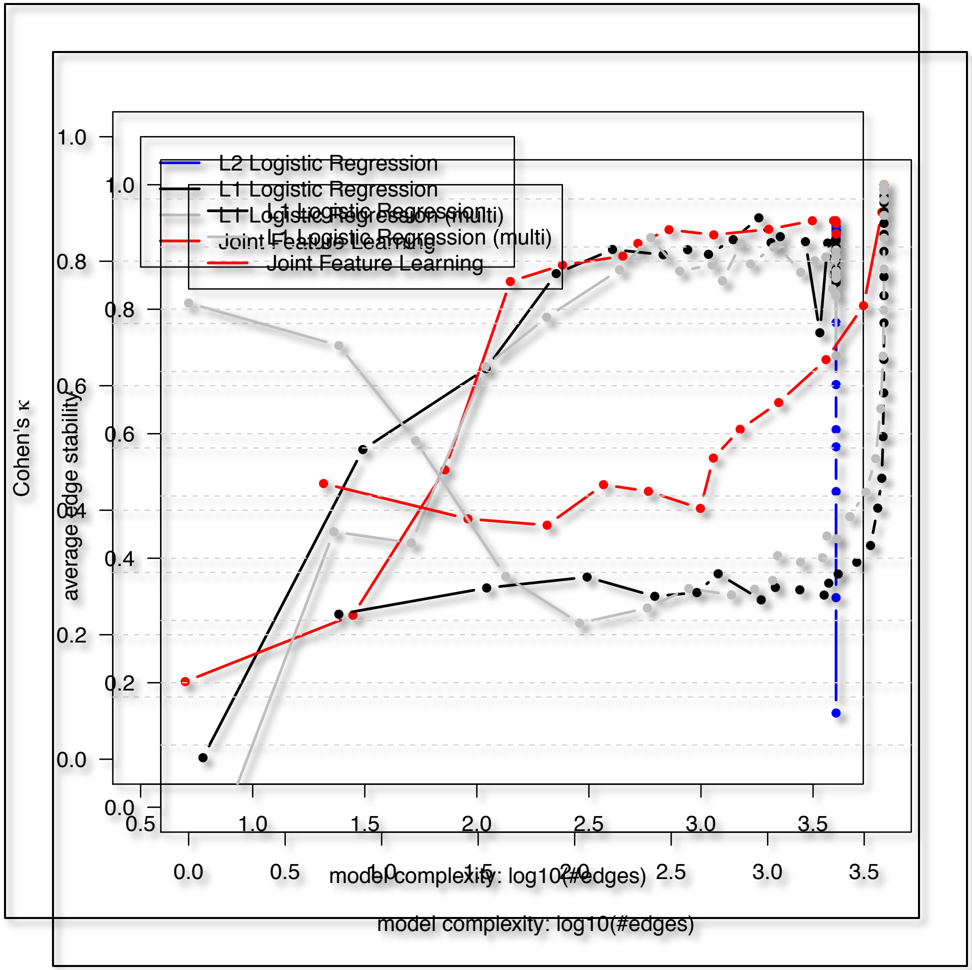


# Predict wakefulness and sleep from rs-fMRI



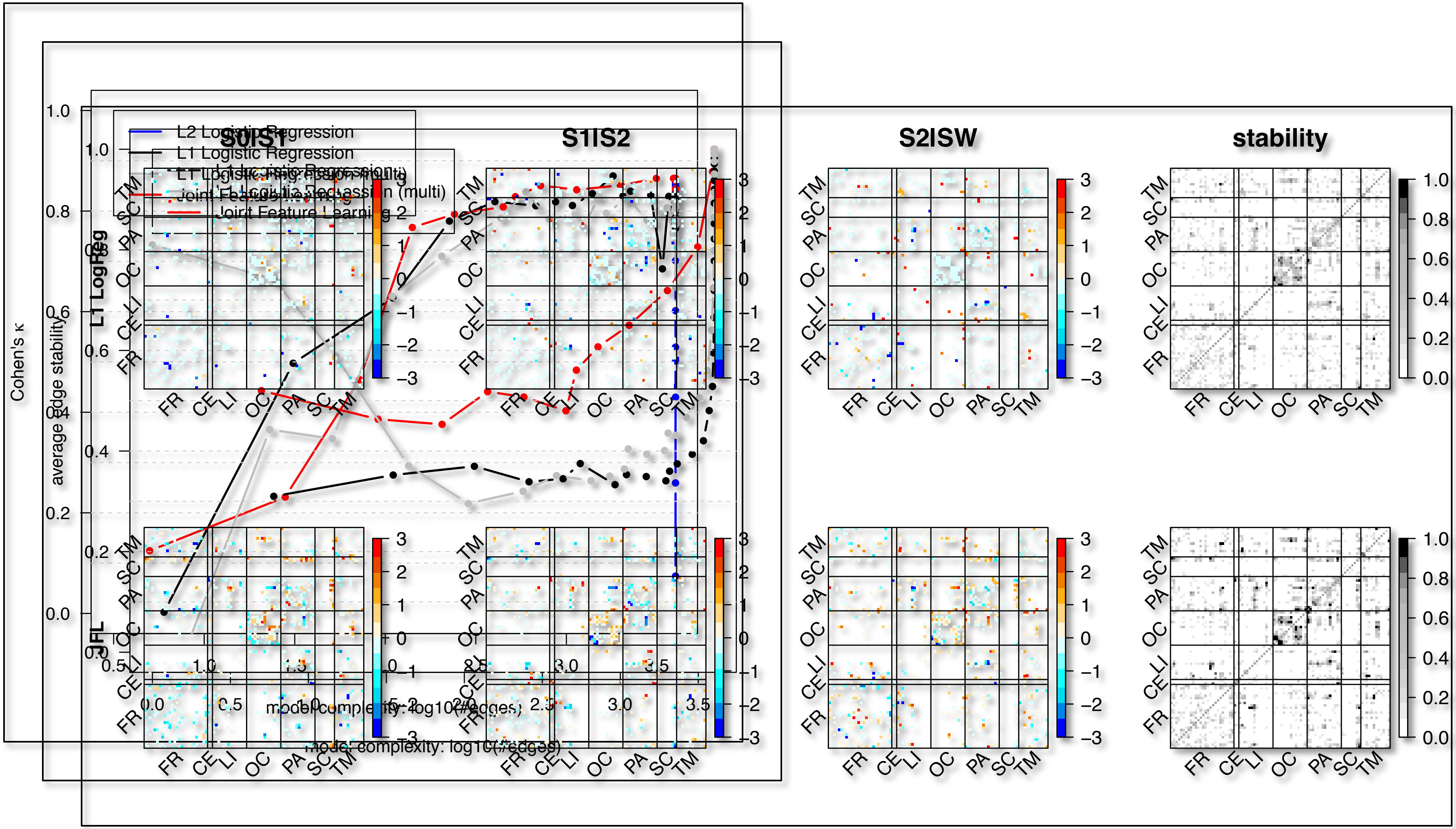

Unchanged performance

# Predict wakefulness and sleep from rs-fMRI




Improved feature stability

# Predict wakefulness and sleep from rs-fMRI



# Conclusions

- MFL results in classifiers with a stable set of features
- No decline in classification accuracy
- Increase in computational cost
- Makes model interpretation a bit simpler
- Works even if tasks are *“not so related”*

# Acknowledgements

Functional Imaging in Neuropsychiatric Disorders Lab
Stanford

STANFORD
SCHOOL OF MEDICINE
Stanford University Medical Center

# *Joint Feature Extraction from Functional Connectivity Graphs with Multi-task Feature Learning*


**André Altmann**, Bernard Ng

Department of Neurology and Neurological Sciences

FIND Lab

altmann@stanford.edu

Stanford, June 11th 2015

# 14 Networks (Shirer et al., 2012)