Multivoxel pattern-based real-time fMRI

Stephen LaConte

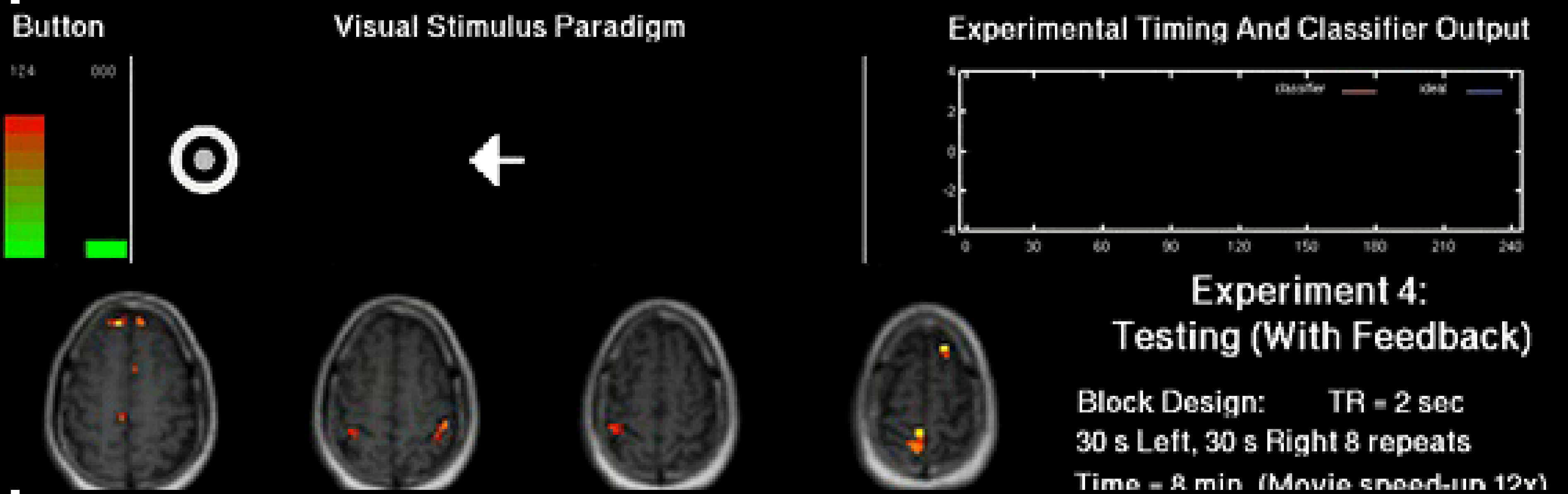
Fralin Biomedical Research Institute
Department of Biomedical Engineering and Mechanics

VirginiaTech 1872

Multivoxel pattern-based real-time fMRI

- Conceptual overview
- Experimental flexibility
- Practicalities and additional resources

Classification in Real Time



LaConte et al. *Hum Brain Mapp* (2007)

Conceptual overview

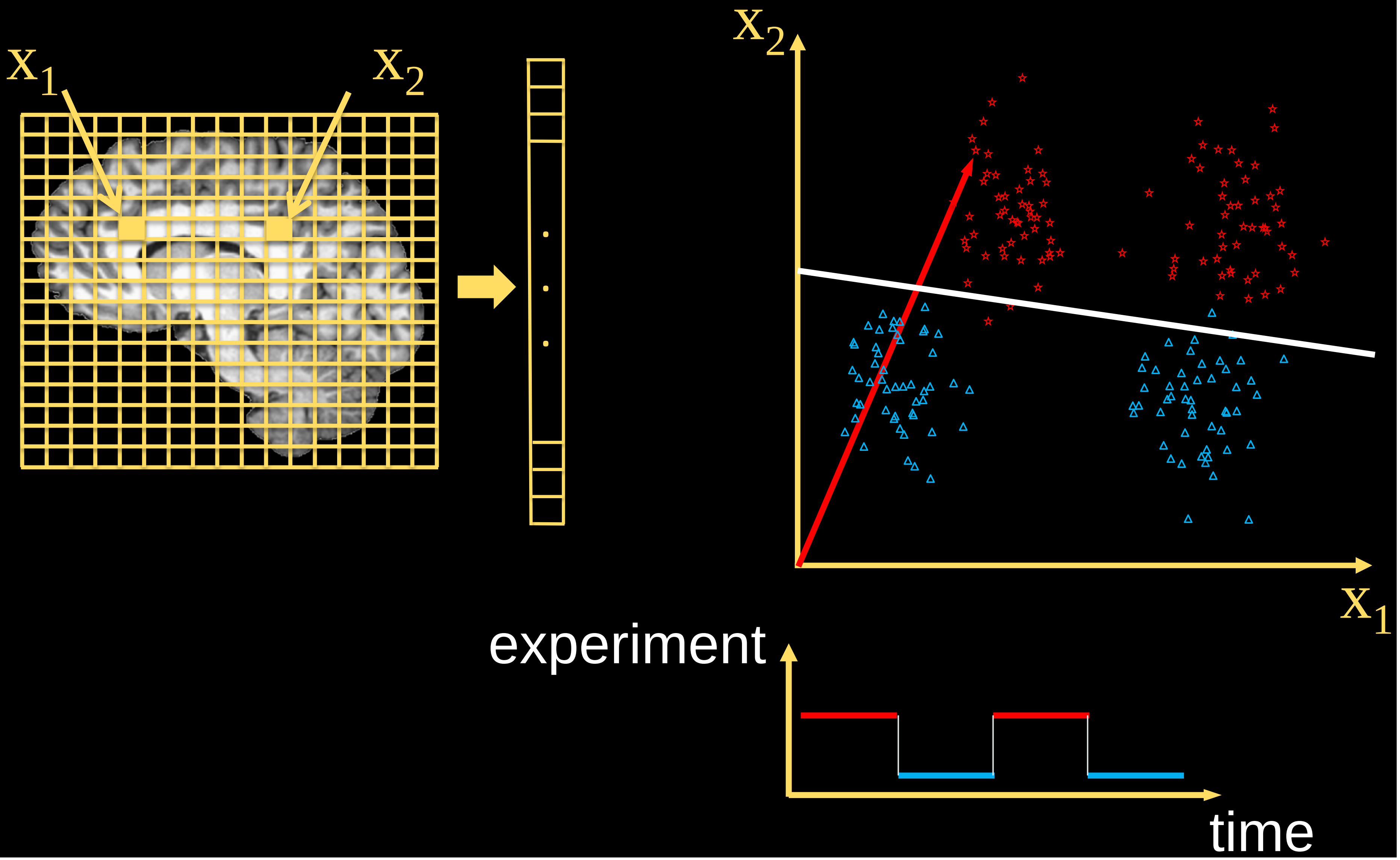
Supervised-learning rtfMRI

- Enables experimental flexibility
- Embodies “MVPA”
  - Complements region-based approaches
  - Enables brain-state decoding
  - Is computationally ideal for rtfMRI

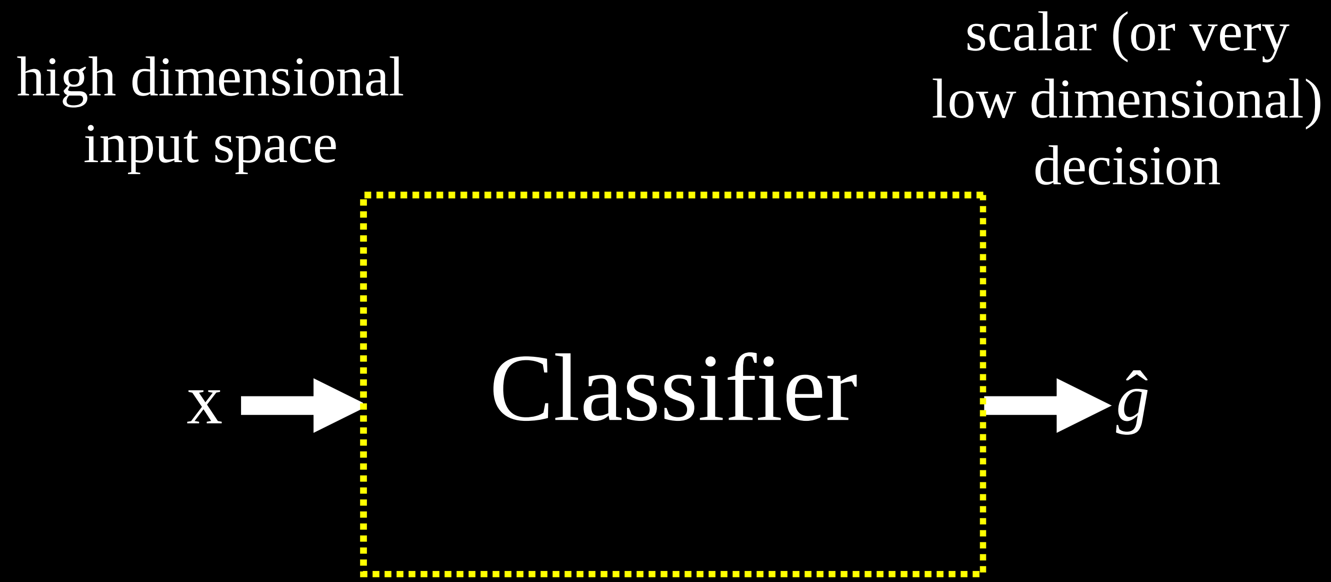
Supervised learning

- Complements univariate approaches
  (Friston, 1995; McIntosh, 1996; Strother, 2002; Moeller and Habeck 2006)
- Early demonstrations
  (Lautrup, 1994; Dehaene, 1998)
- Methodology and validation
  (Strother, 2002; LaConte, 2003; Mitchell 2004)
- Representation of
  different classes of stimuli
  (Haxby, 2001; Cox and Savoy, 2003;
  Haynes & Rees, 2005; Kamitani & Tong 2005)
- Detecting and tracking
  cognitive states
  (Polyn, 2005)
- Natural representation
  for real-time fMRI
  (LaConte, 2007; Shibata, 2011; deBettencourt, 2015)

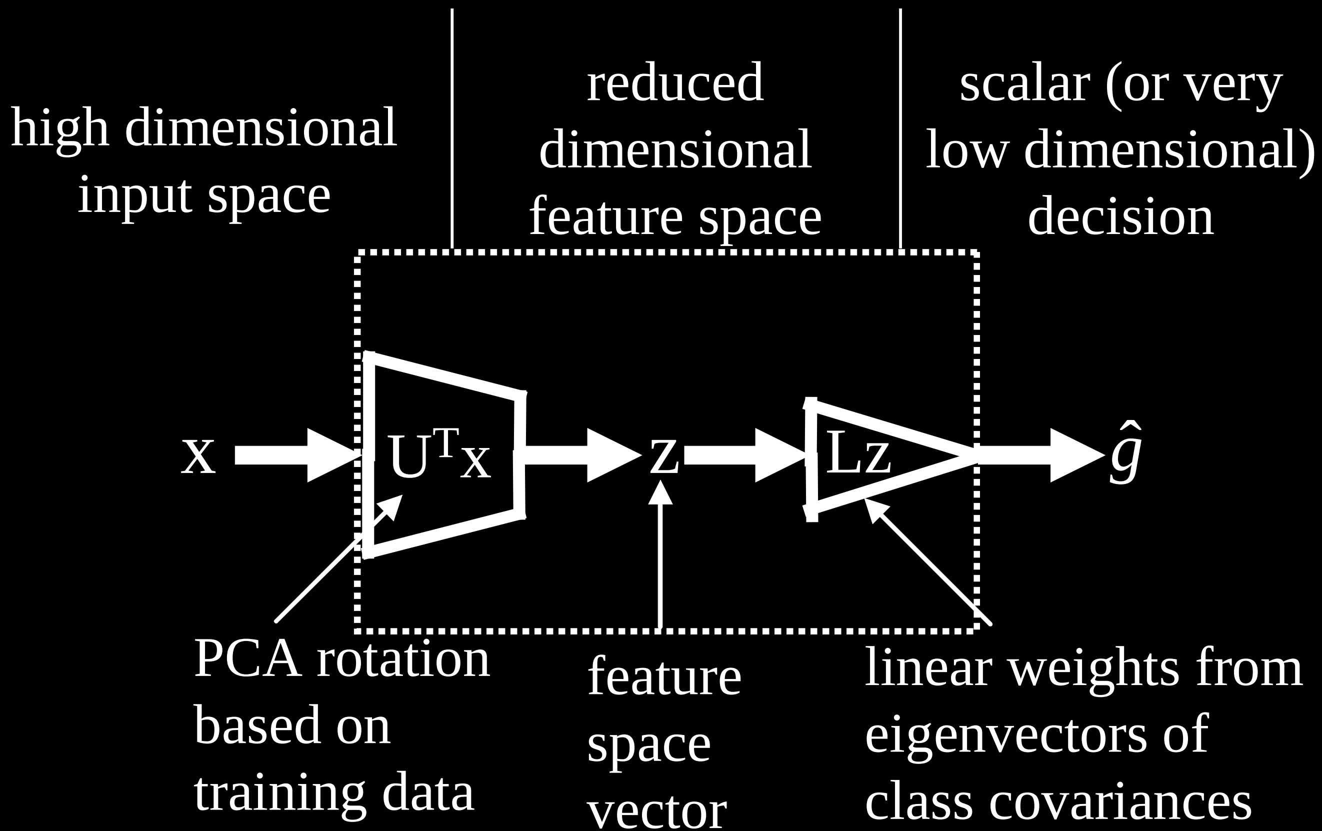
Brain State Classification



Classification



Linear Discriminant Analysis



Linear Discriminant Analysis

Data Matrix: $\mathbf{X}$ (voxels ↓, time →)

PCA via SVD:

$$\mathbf{U}^{\mathrm{T}}\mathbf{X} = \mathbf{\Lambda}\mathbf{V}^{\mathrm{T}} = \mathbf{Z}$$

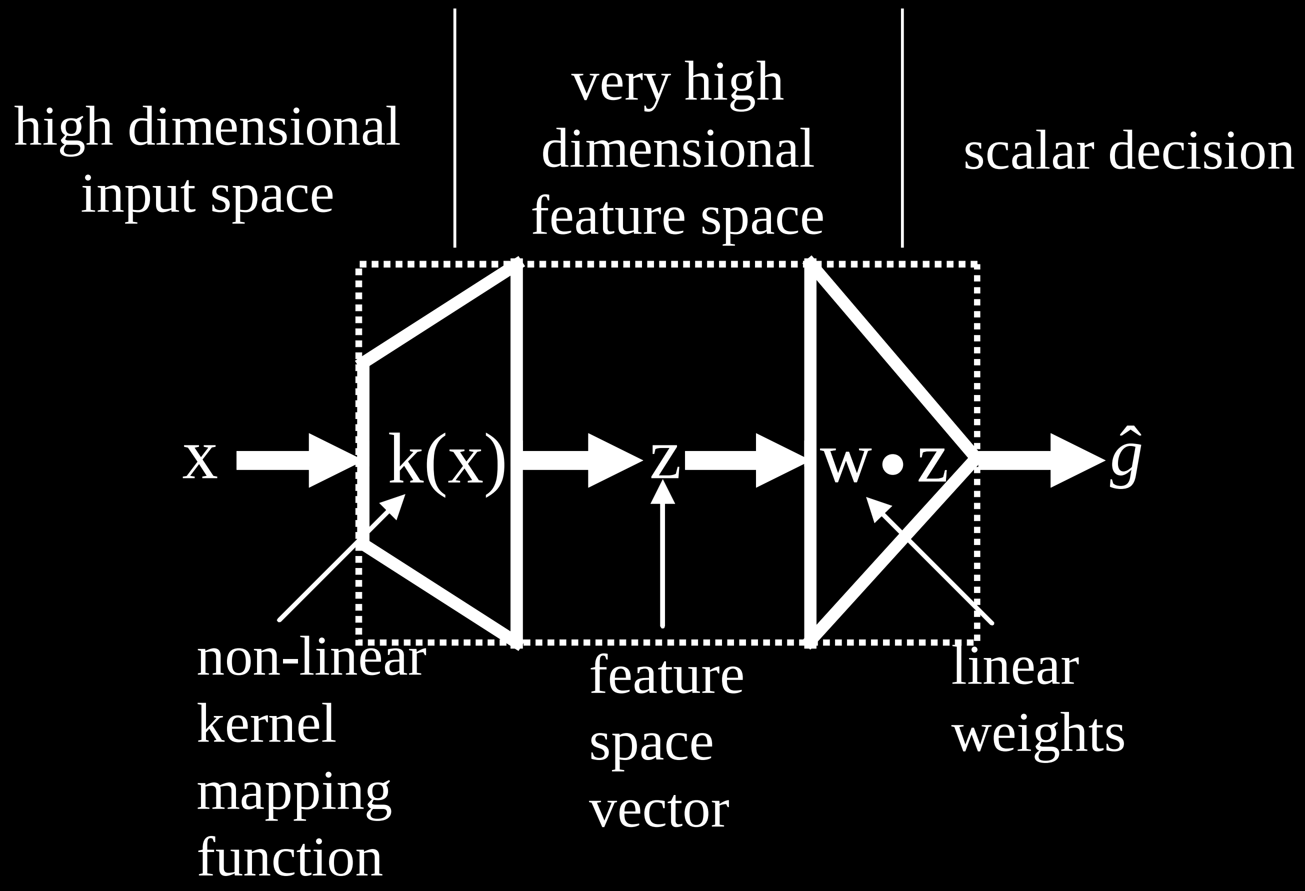
Truncate Q (model complexity)

CVA:

$$\mathbf{C} = \mathbf{L}\mathbf{Z}^{*} = \mathbf{L}\mathbf{U}^{\mathrm{T}*}\mathbf{X}$$

Columns of L are determined by the eigenvectors of $W^{-1}B$. W is the within class variance and B the between class variance, and both are obtained from Z.

Support Vector Machine



SVM

$$D(\vec{z}_t) = (\overrightarrow{w} \cdot \vec{z}_t) + w_0)$$

minimize

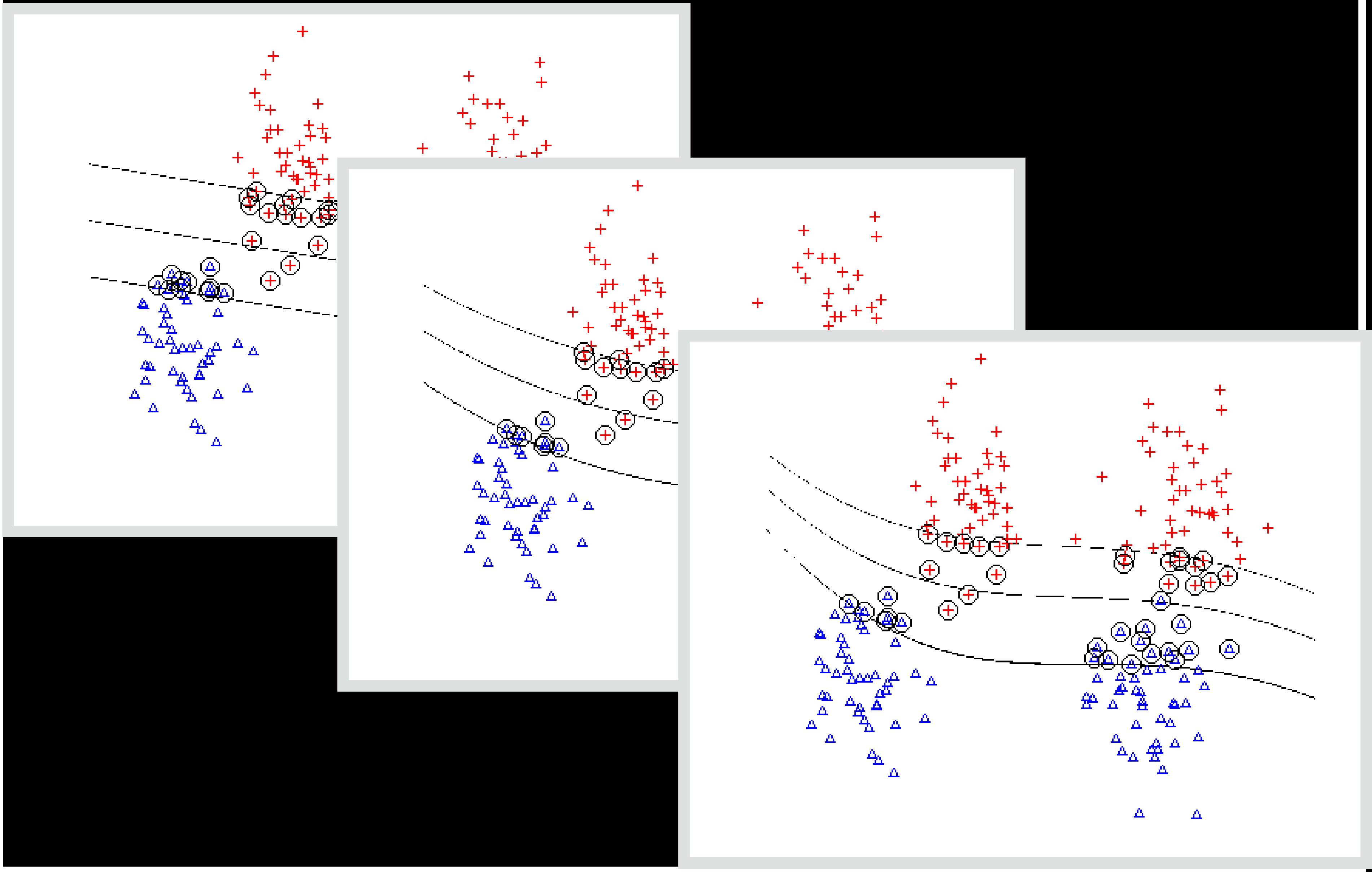
$$\frac{C}{T}\sum_{t=1}^{T}\xi_t + \frac{1}{2}\|\overrightarrow{w}\|^2$$

This term allows some training errors.

This term favors the widest possible margin,

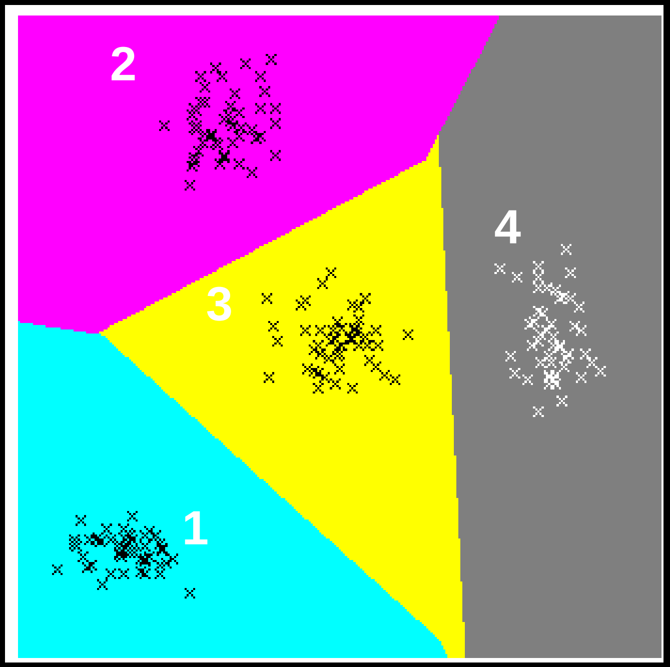
C = infinity is hard margin SVM (as apposed to soft margin) because it does not allow any training errors

Nonlinear Decision Boundary

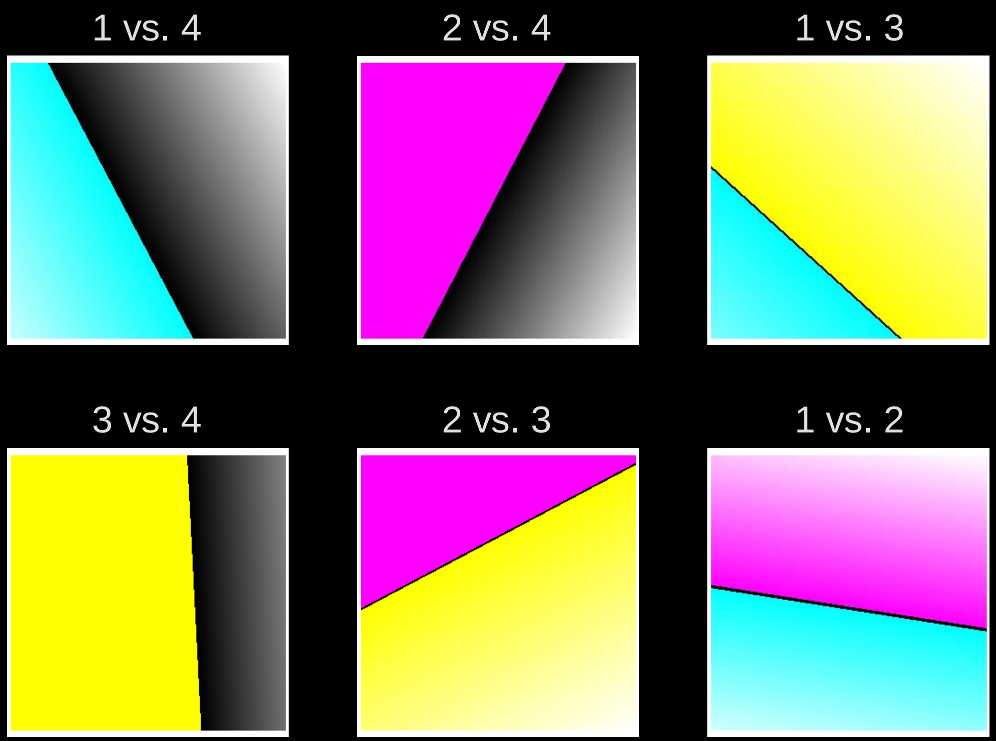


Multi-class

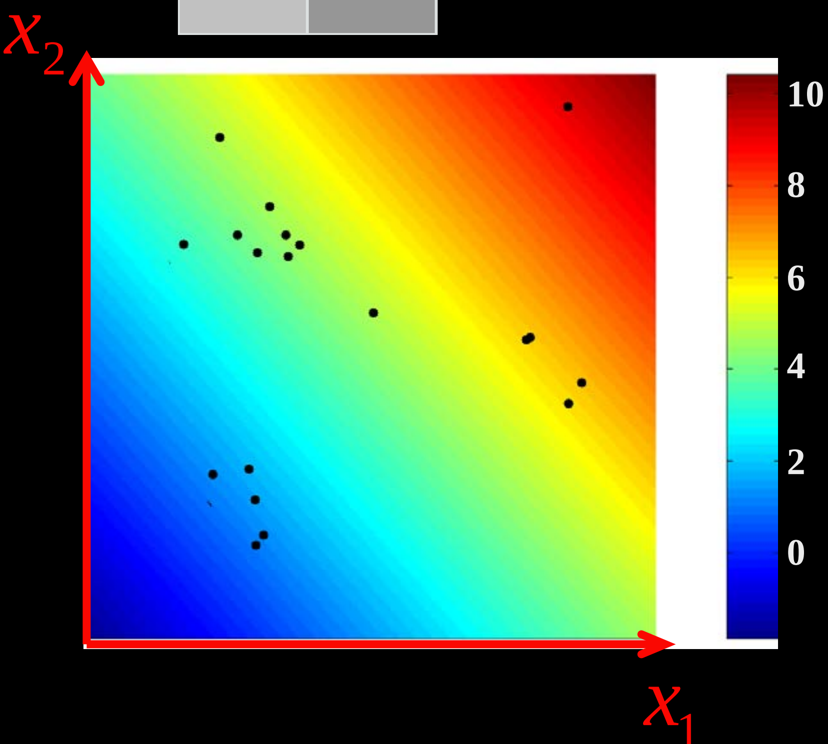
4-Class Model

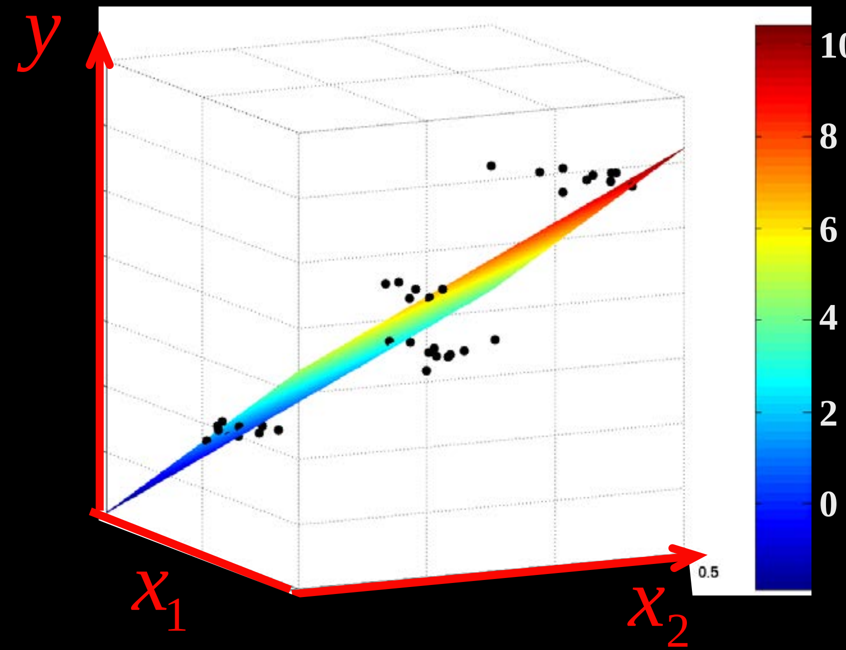


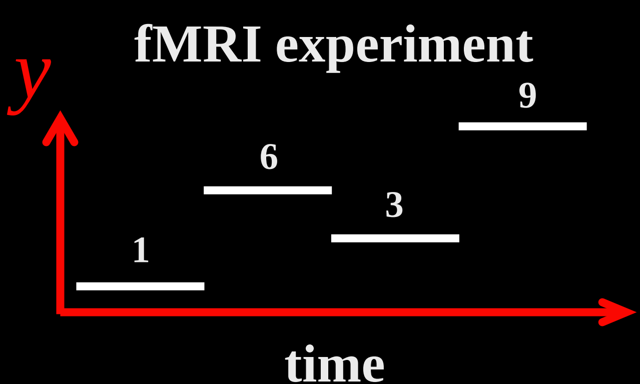
Individual 2-class models



Temporal Regression



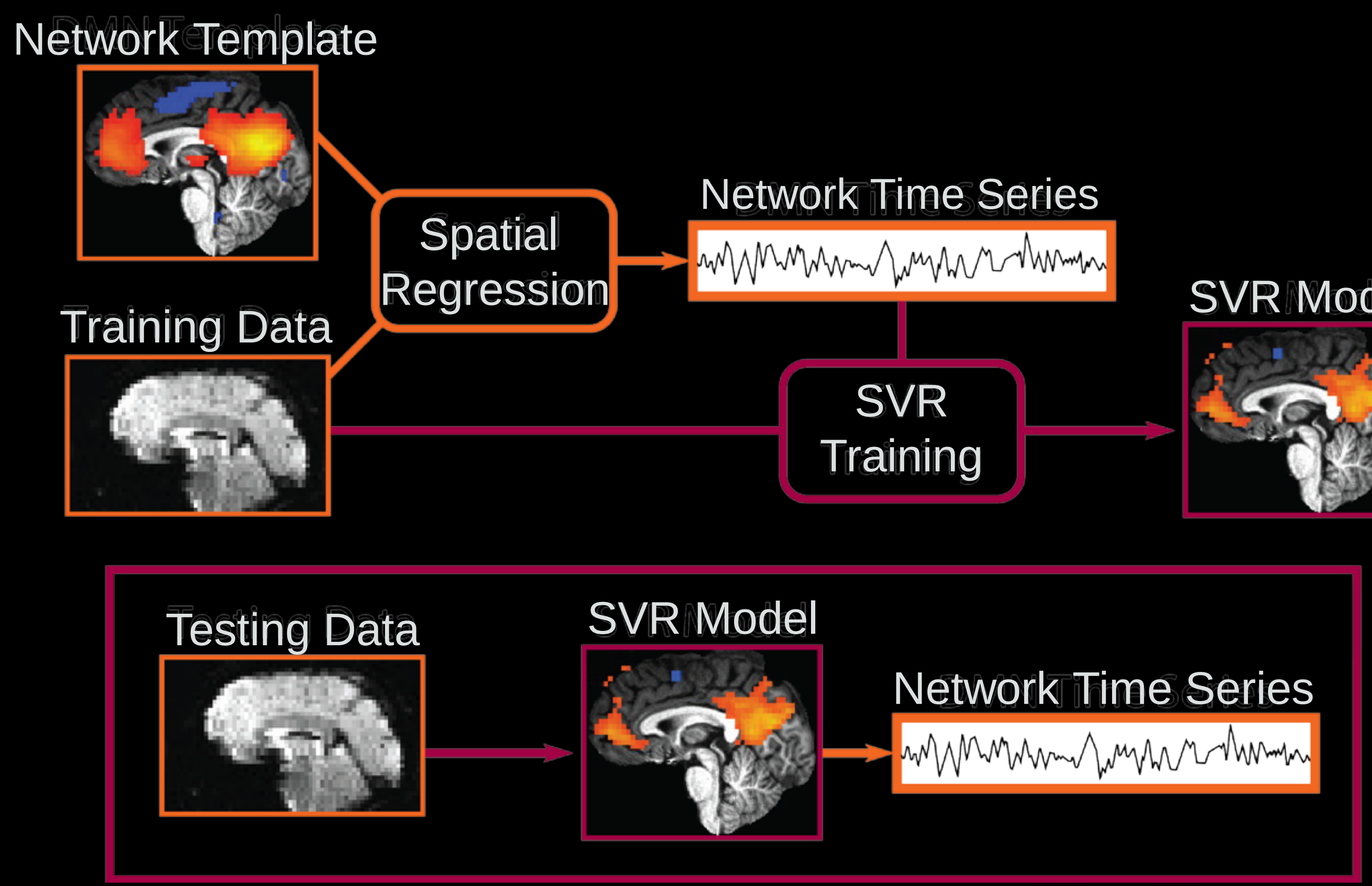




$$f(x) = x^T \beta + \beta_0$$

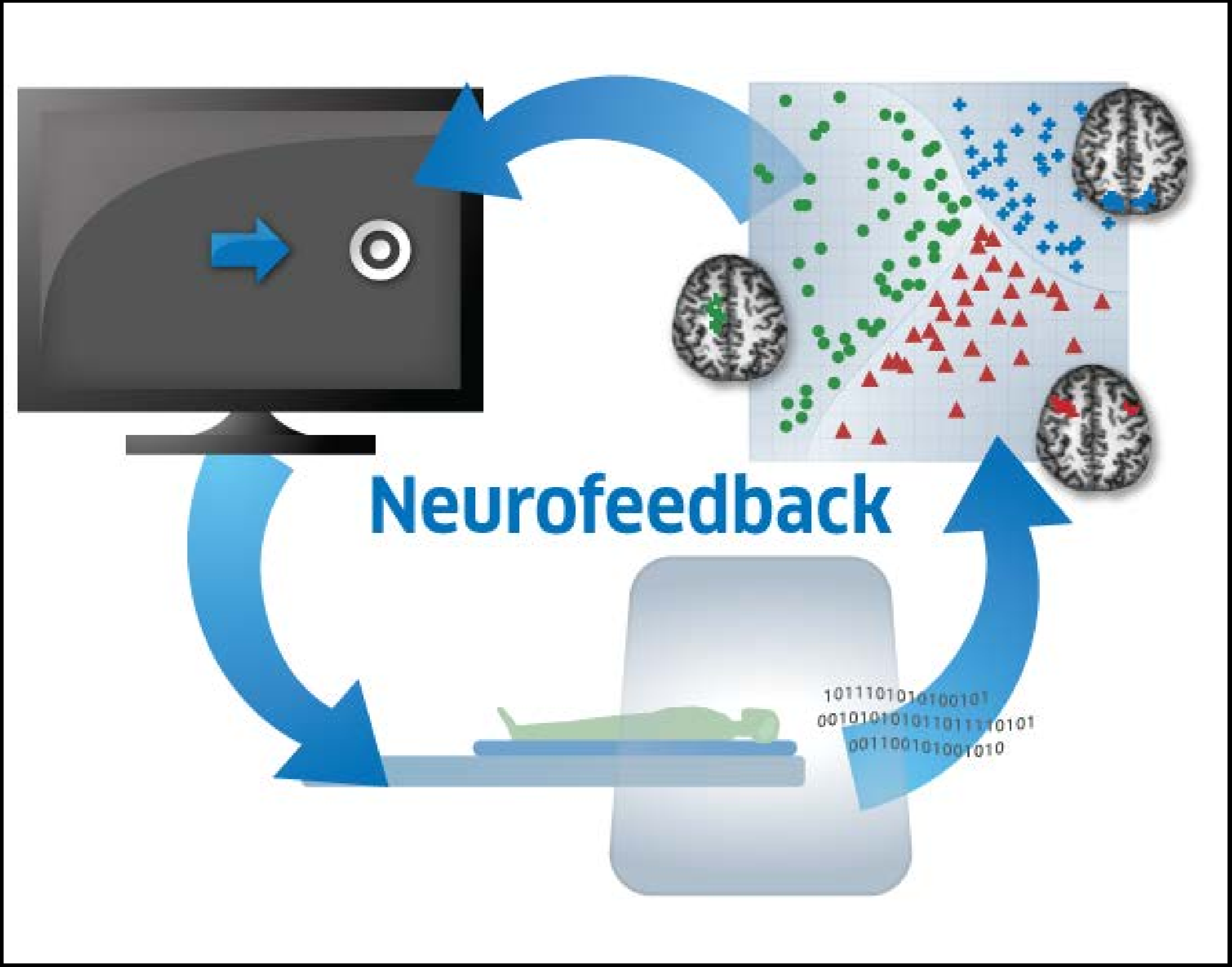
$$L(y, f(x))$$

Predicting Network Time Series

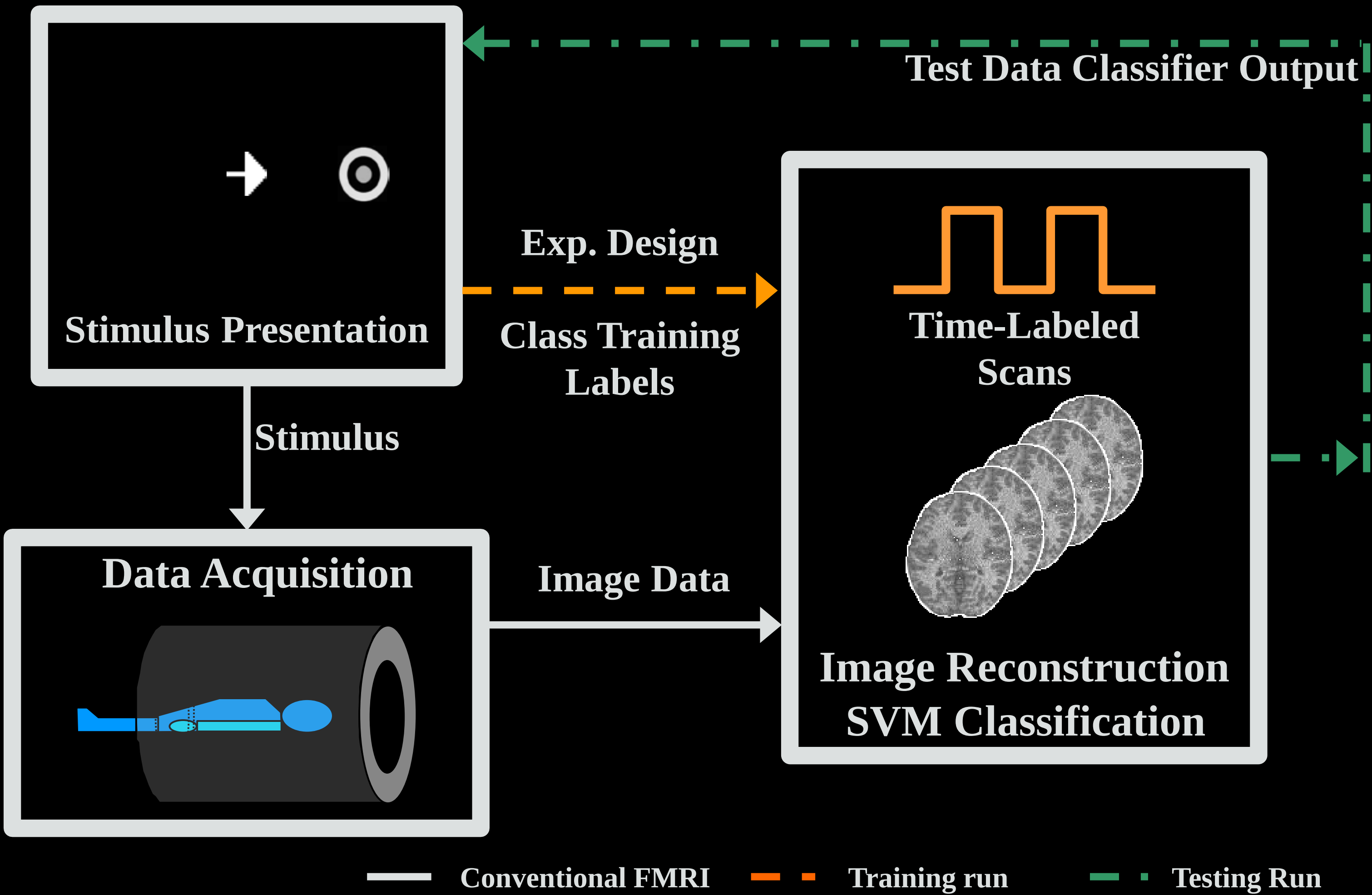


Craddock, R.C. et al. OHBM 2012.

Pattern-based rtfMRI

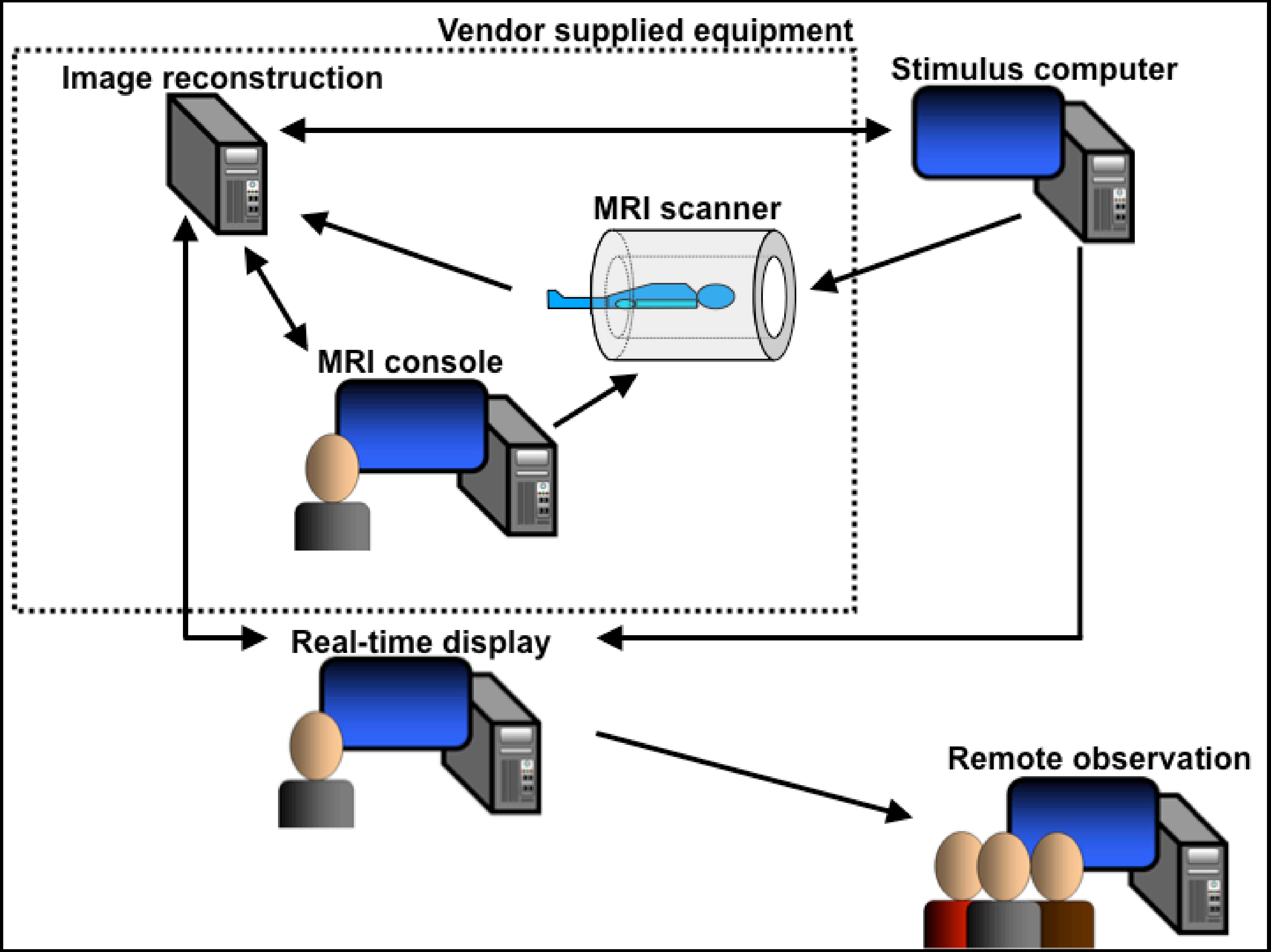


Pattern-based rtfMRI



LaConte, et al. (2007) Hum Brain Mapp. 28: 1033-1044

Network configuration

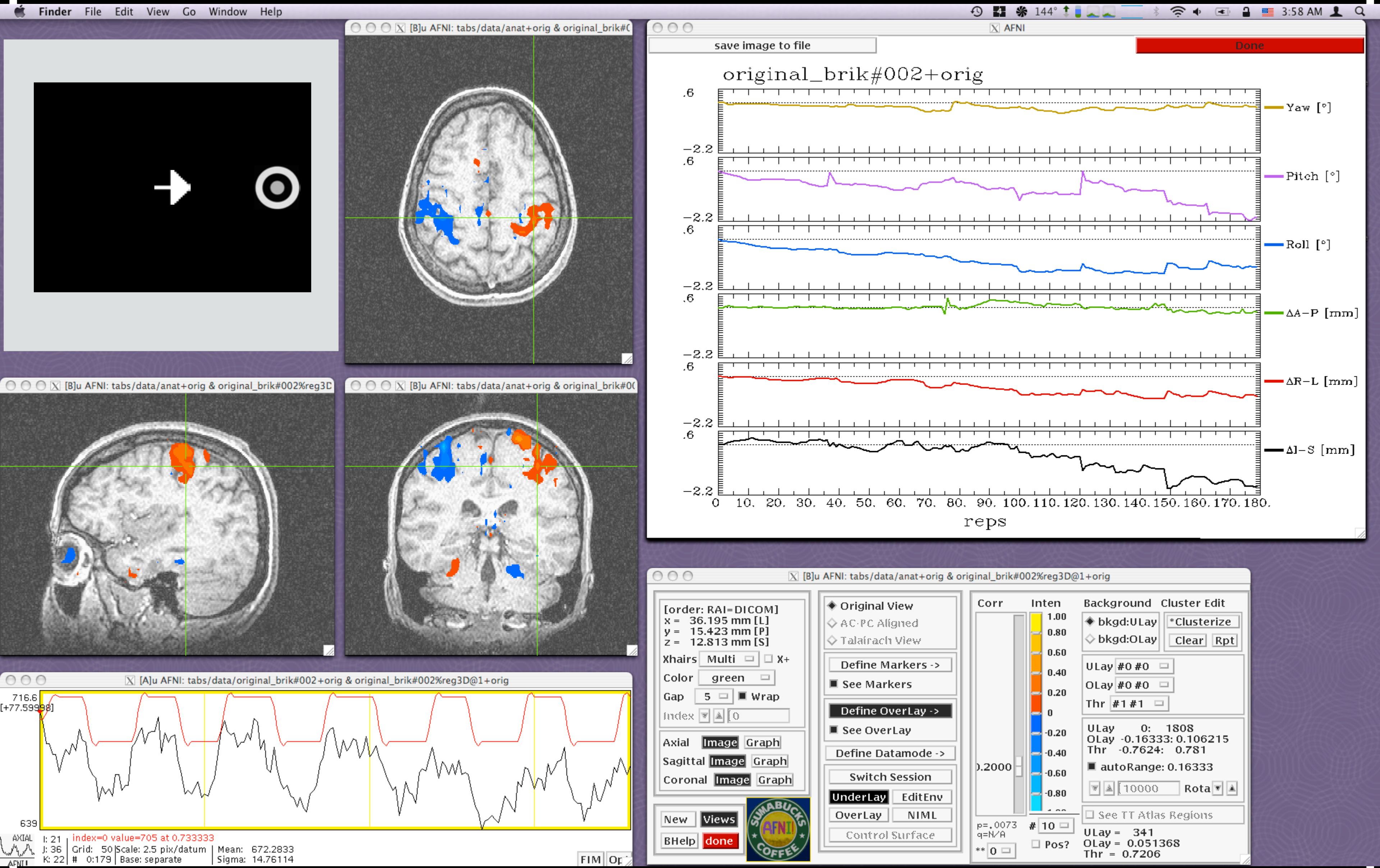


LaConte, *NeuroImage* (2011)

Stimulus seen by volunteer

Updated fMRI results

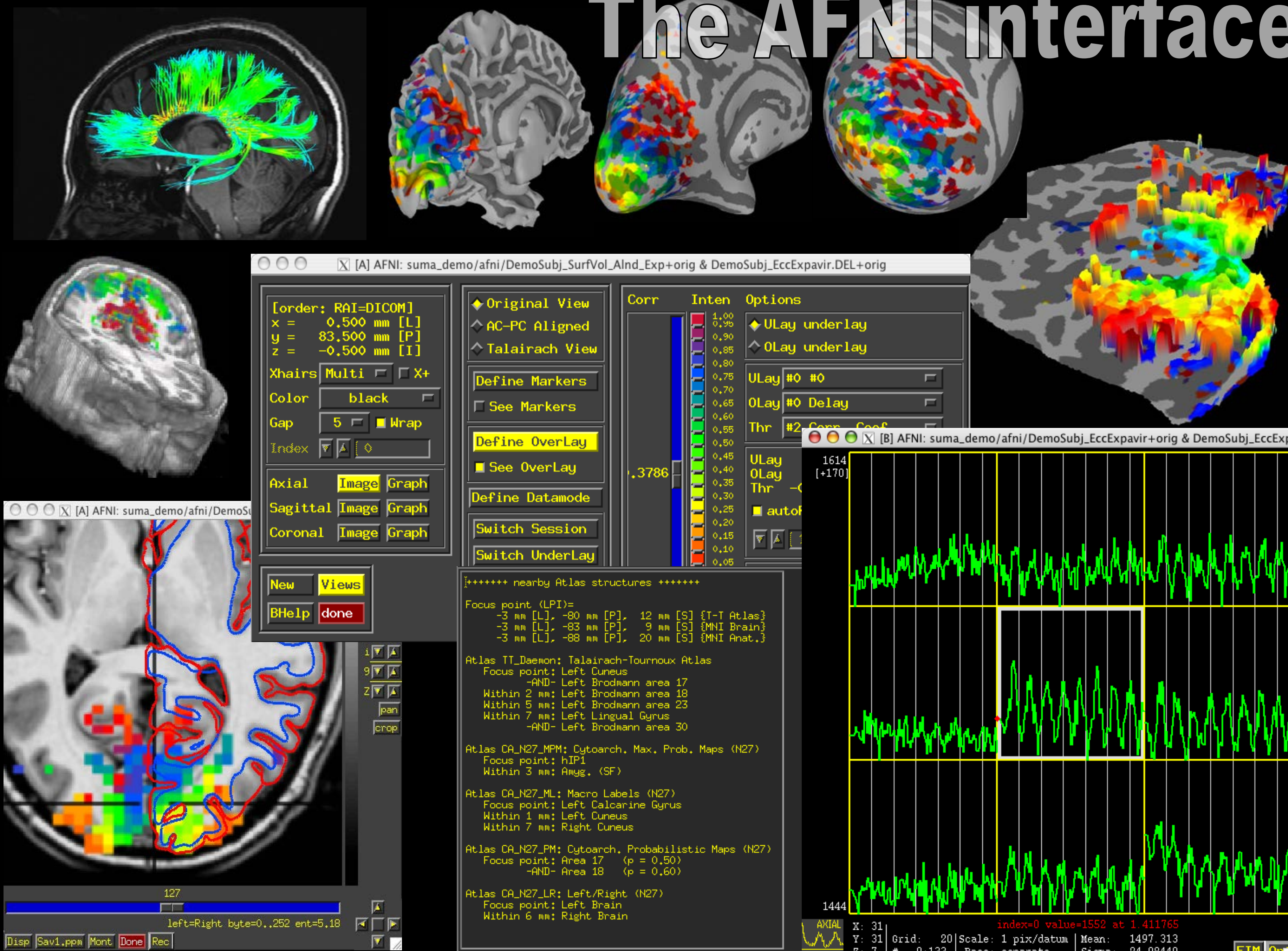
Motion tracking and correction



Intensity (brightness) of a single pixel, changing during stimulus conditions

Controller interface for some display parameters

The AFNI interface

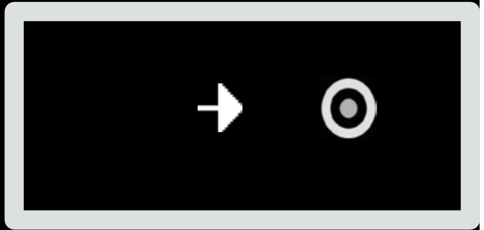


Slide provided by Ziad Saad

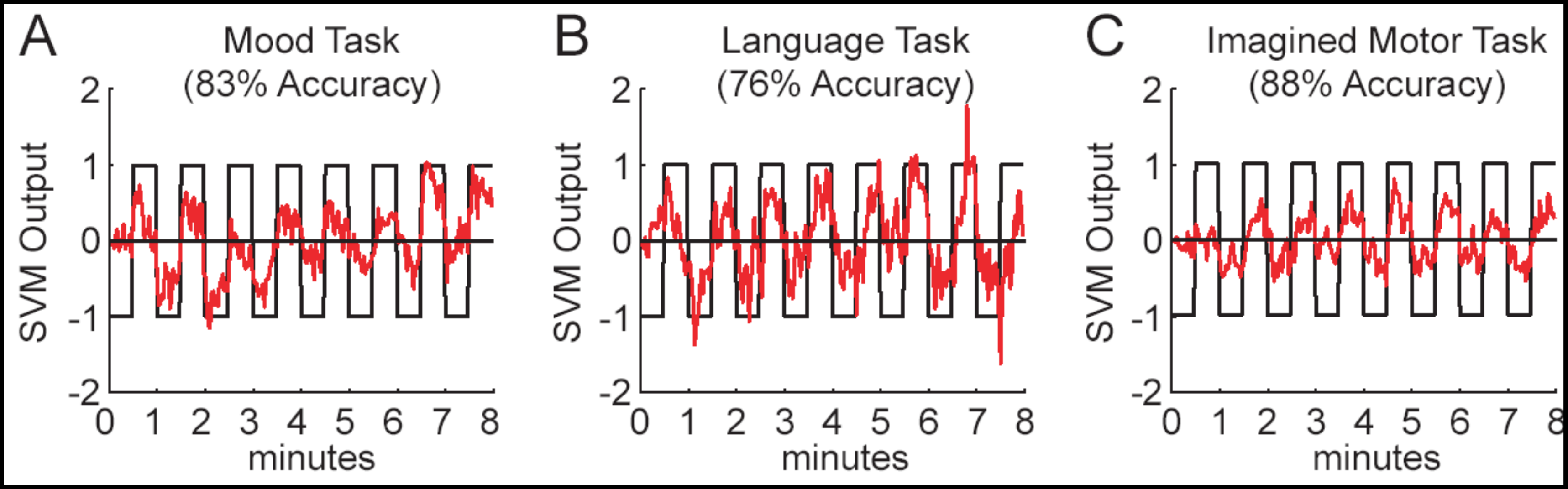
Multivoxel pattern-based real-time fMRI

- Conceptual overview
- Experimental flexibility
- Practicalities and additional resources

Flexibility of brain state classification



With the exact same experimental setup (different instructions), subjects can learn to move the arrow

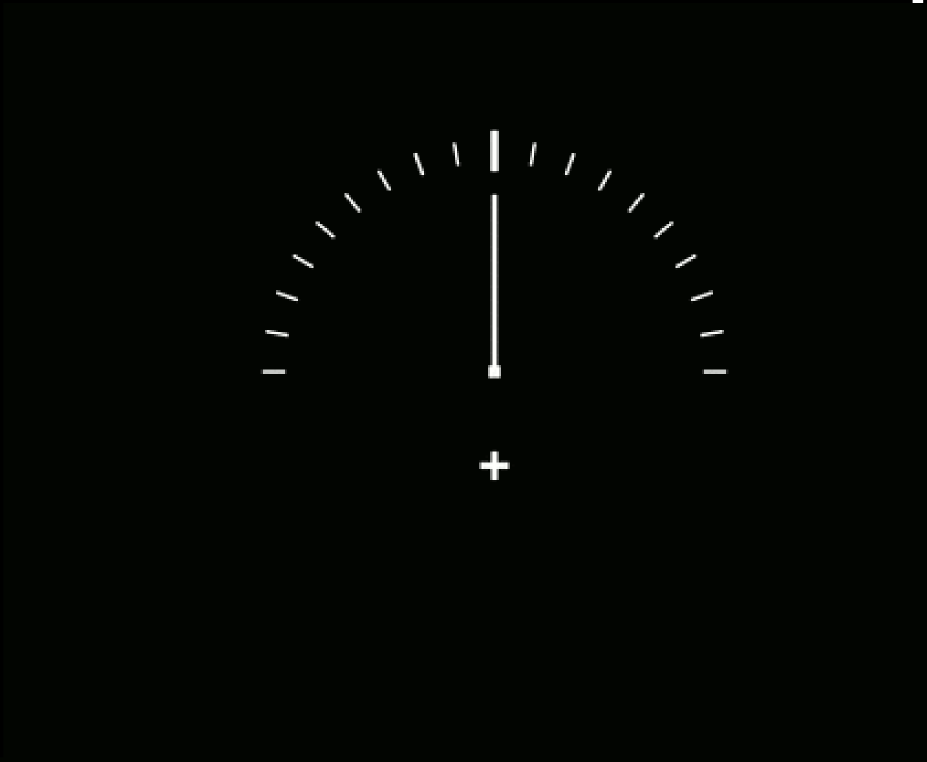


LaConte, et al. (2007) Hum Brain Mapp

Experimental flexibility:
Task appropriate interfaces

Cigarette craving

Covert counting rate



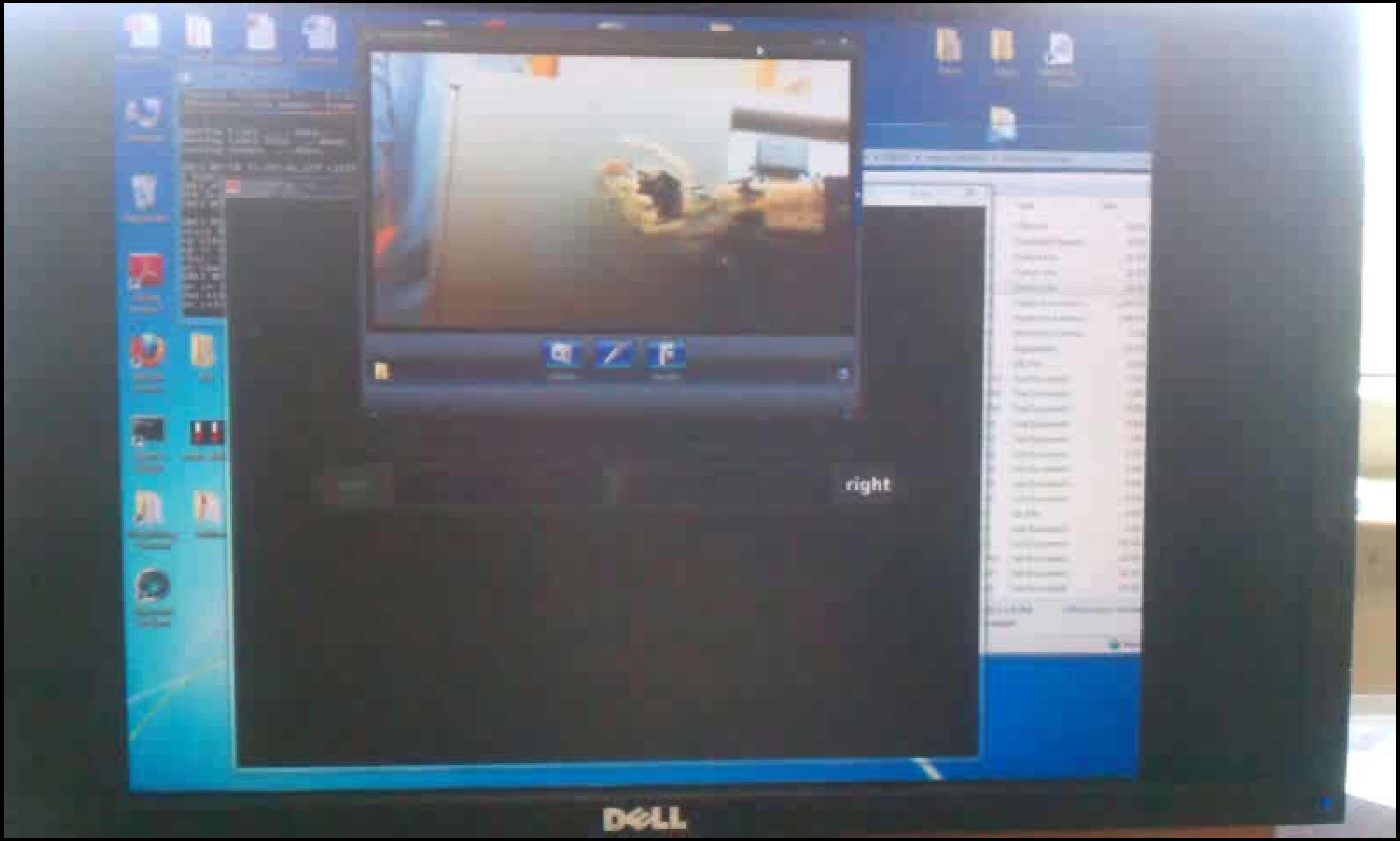
In collaboration with Pearl Chiu

Experimental flexibility:
Support vector regression of RSNs

neurofeedback

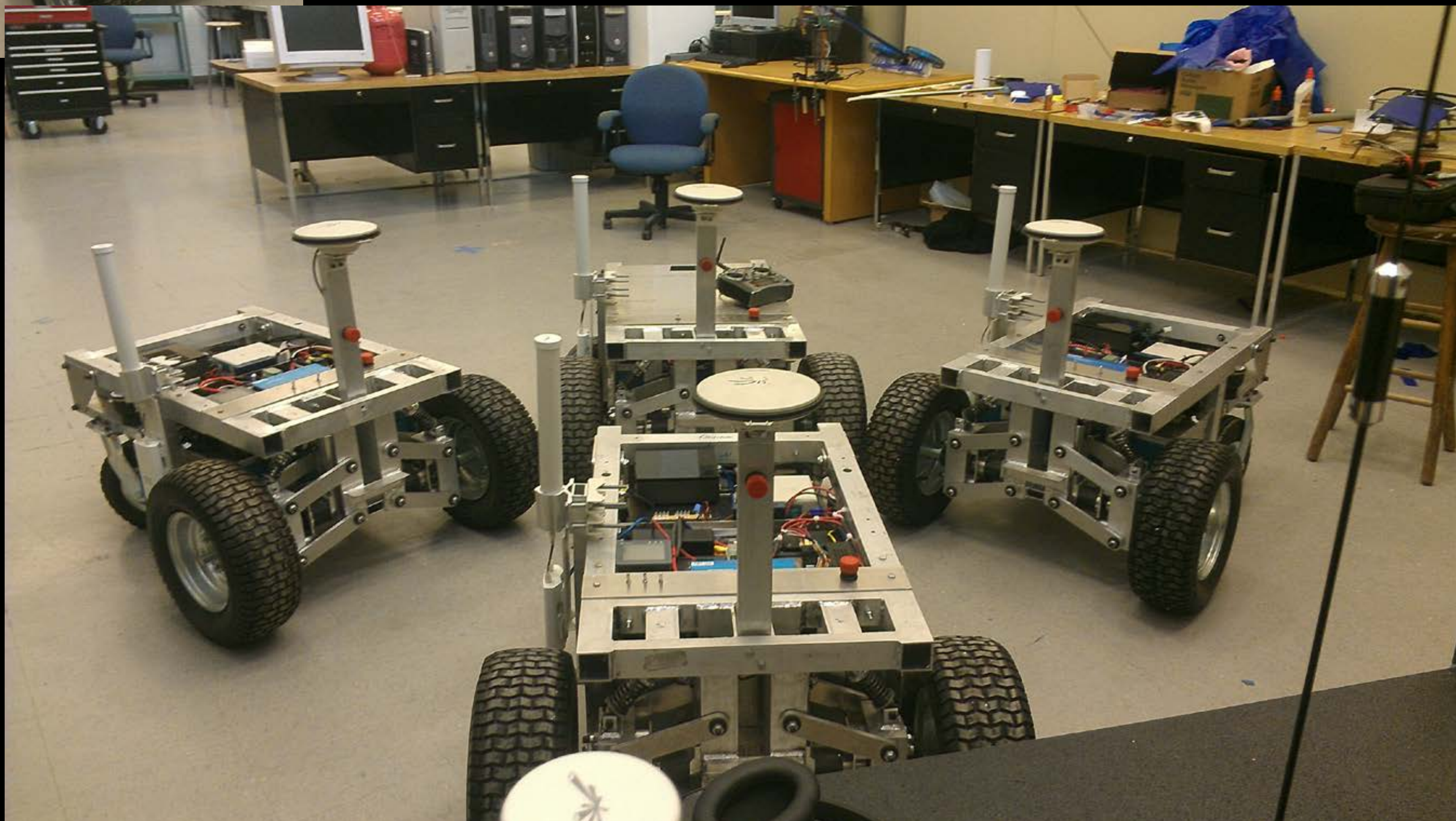
controling events

Experimental flexibility: Brain Machine Interfaces

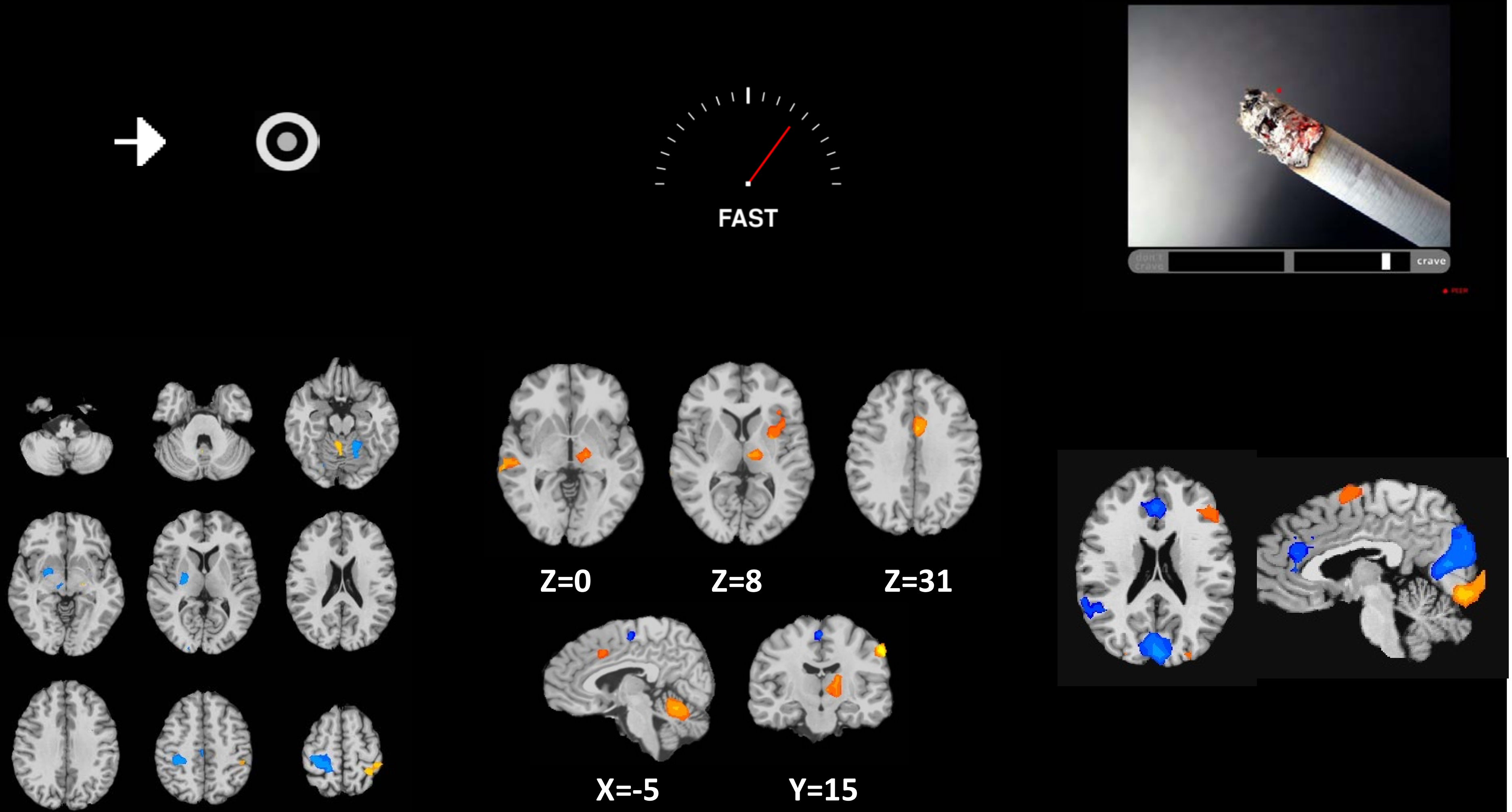


with Shashank Priya and Read Montague

Experimental flexibility: Brain Machine Interfaces



Support Vector Machine Maps of Real-Time Tasks

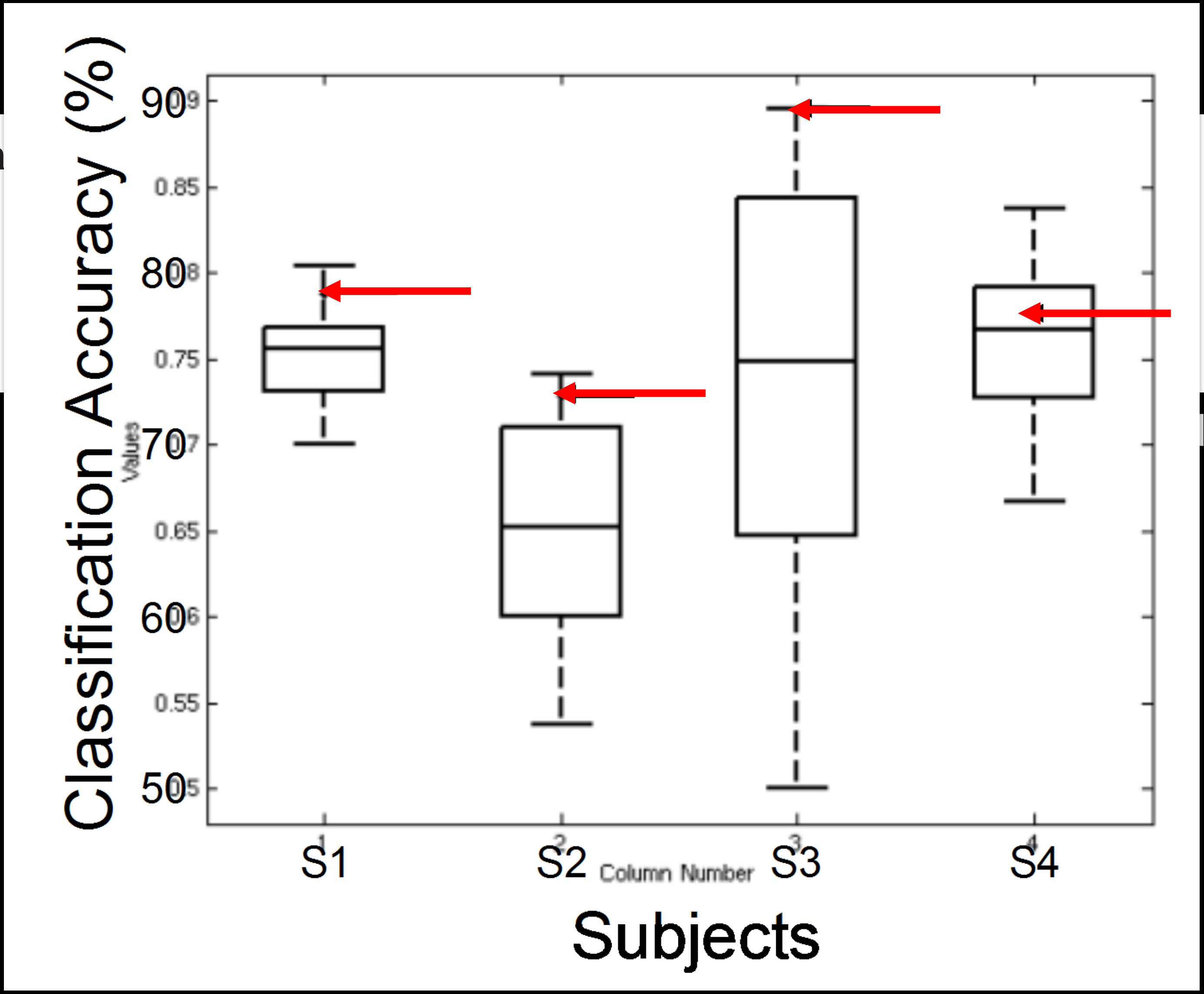


Right vs. Left Tapping **Fast vs. Slow Counting** **Crave vs. Don't Crave**

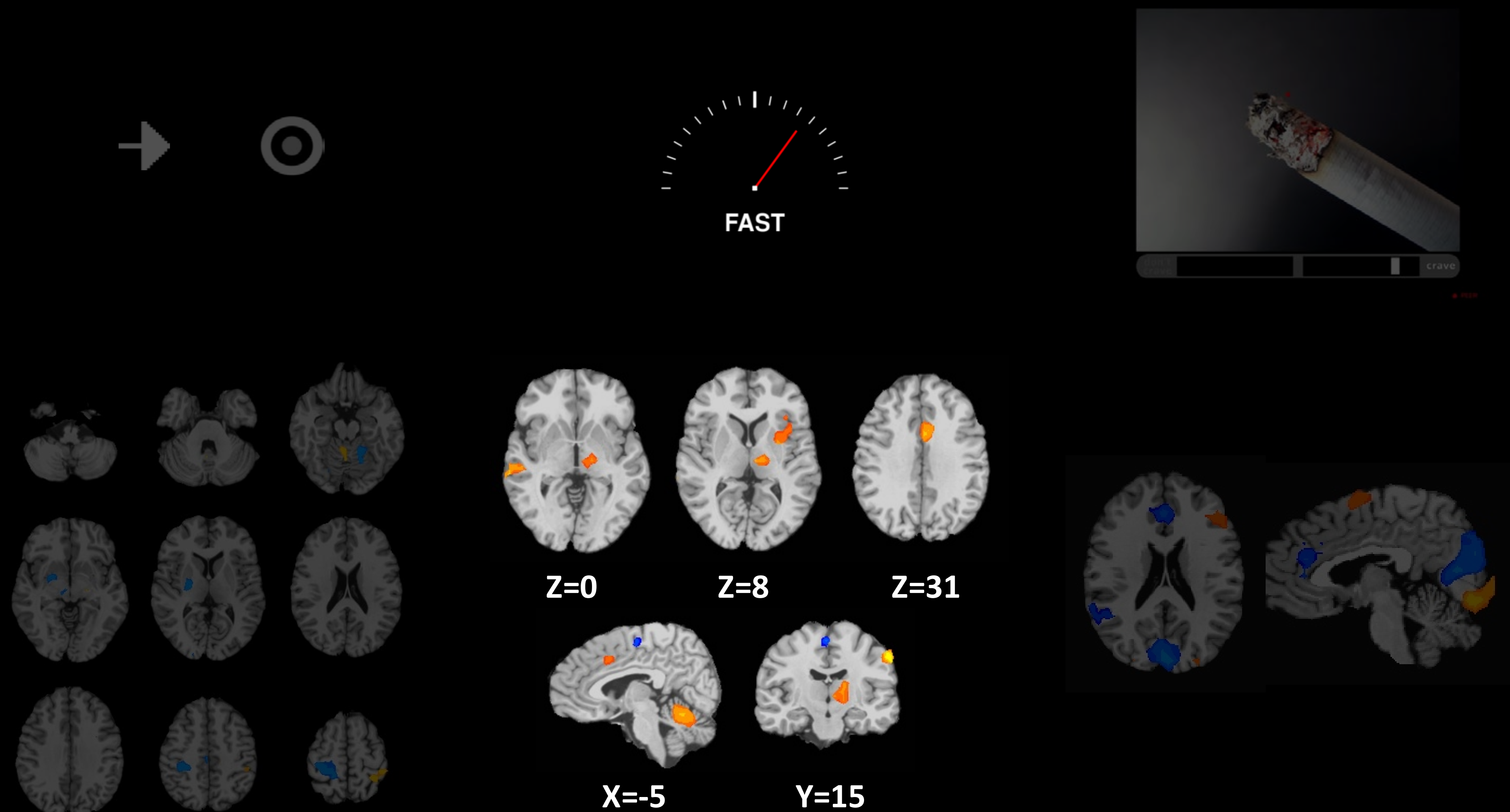
Do feedback runs differ from no-feedback runs?

[1]We h

, 2007)



Support Vector Machine Maps of Real-Time Tasks

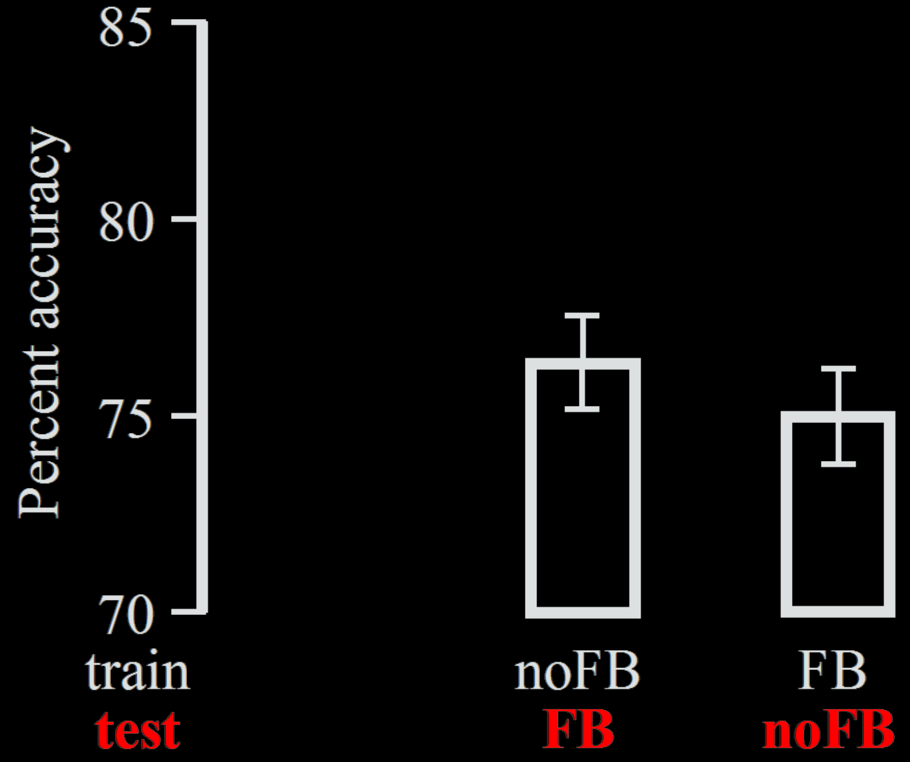


Right vs. Left Tapping

Fast vs. Slow Counting

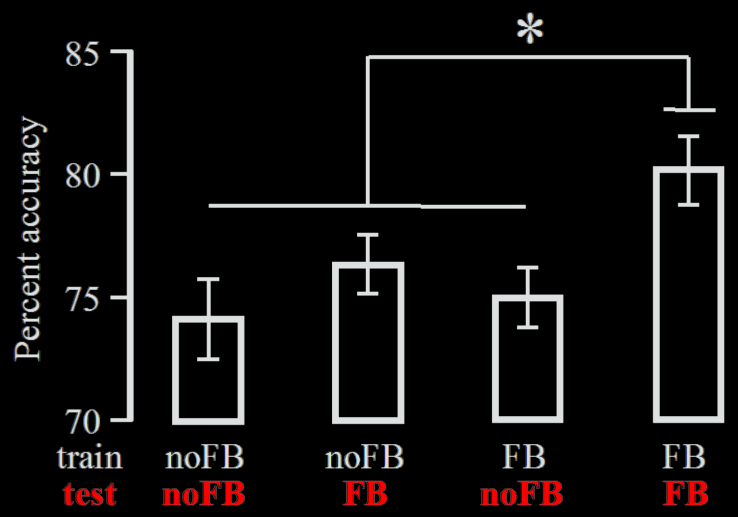
Crave vs. Don’t Crave

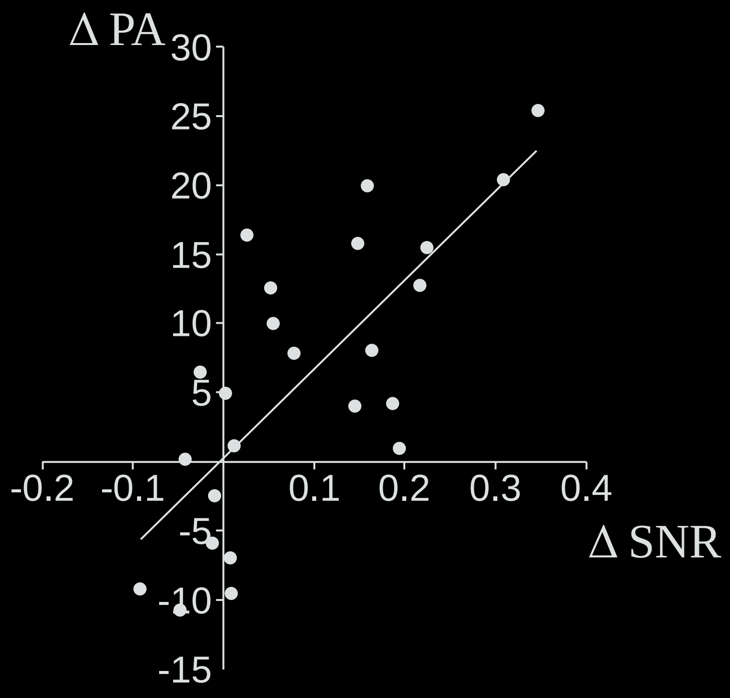
Speech: Covert counting
Classification accuracy

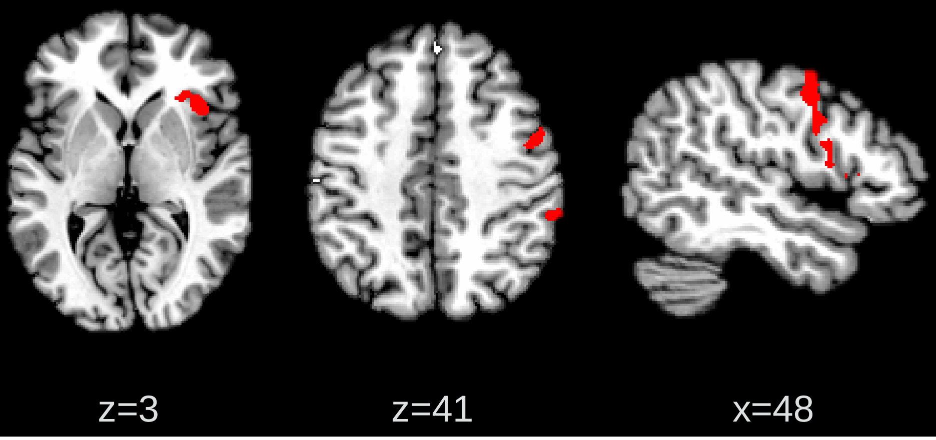


Papageorgiou, et al.
PNAS, 2013.

Speech: Covert counting

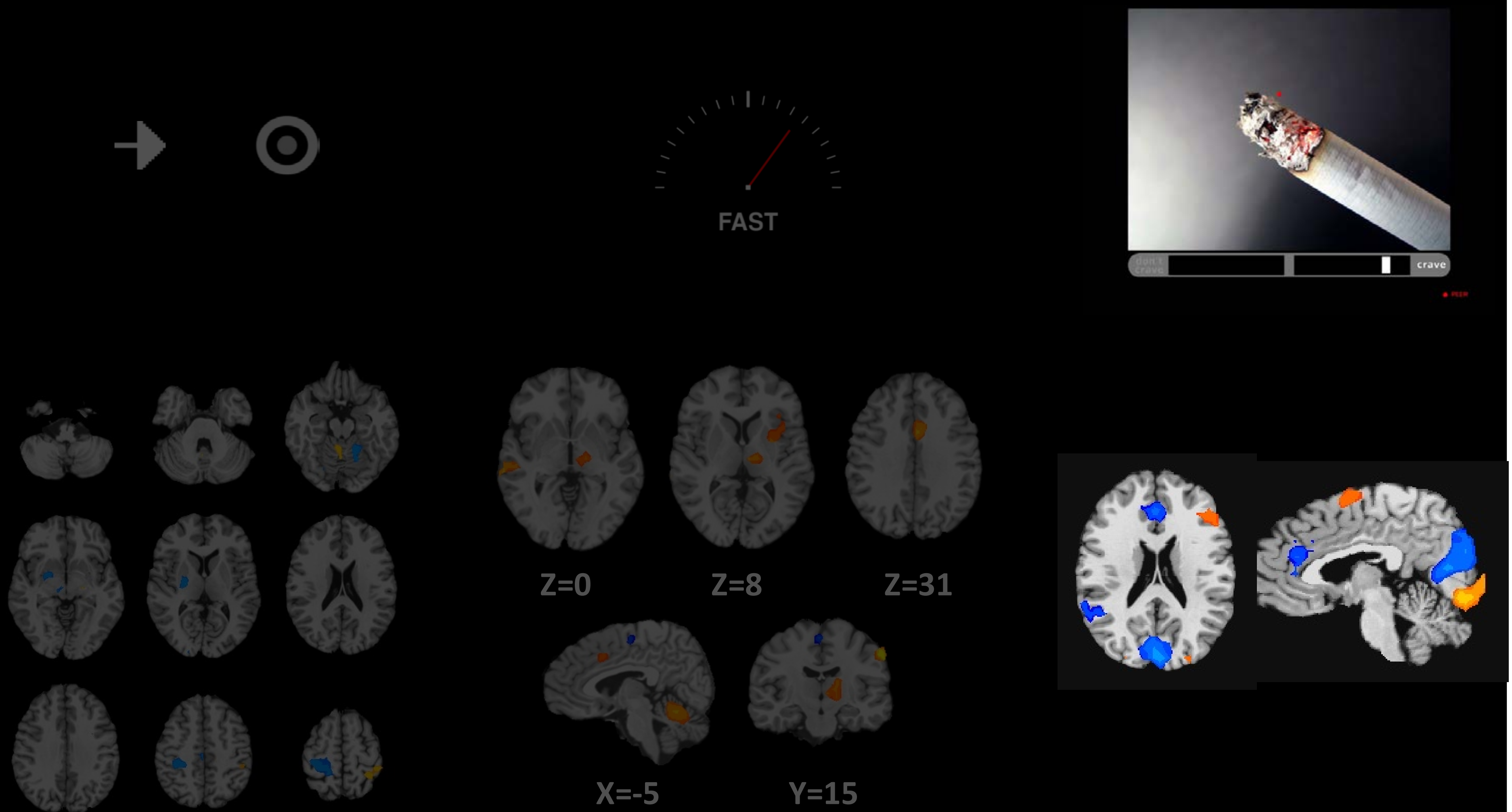






Papageorgiou, et al.
PNAS, 2013.

Support Vector Machine Maps of Real-Time Tasks

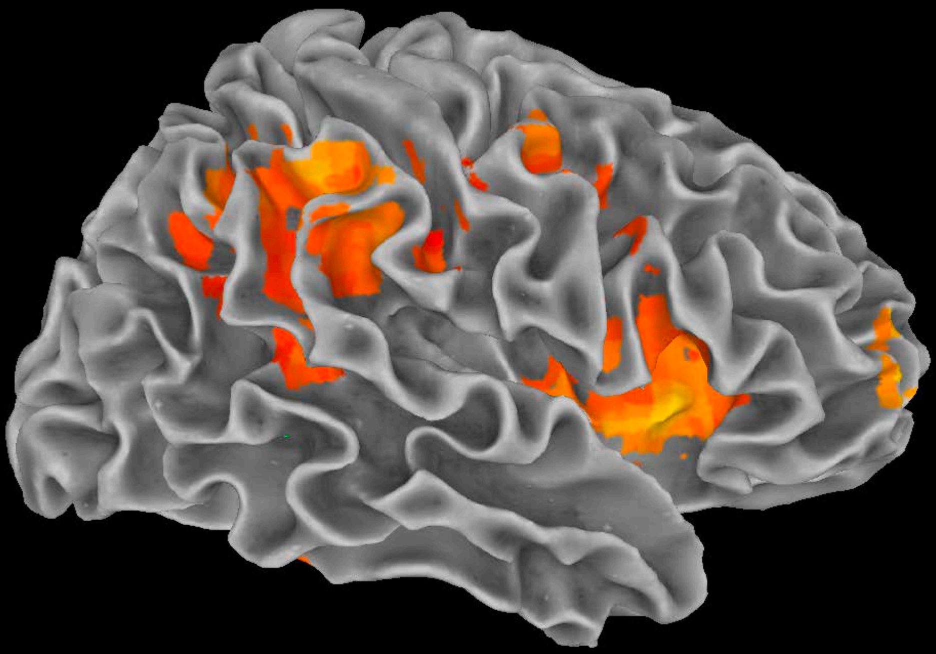


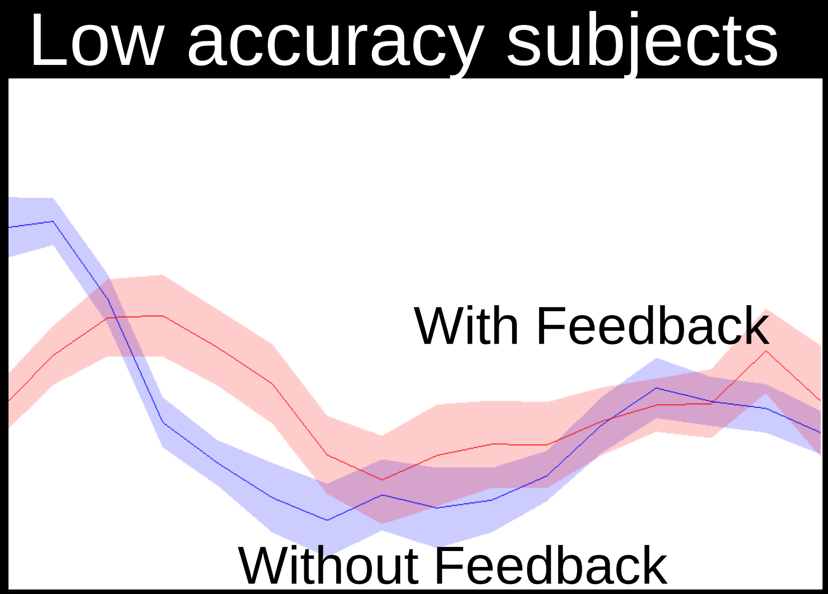
Right vs. Left Tapping

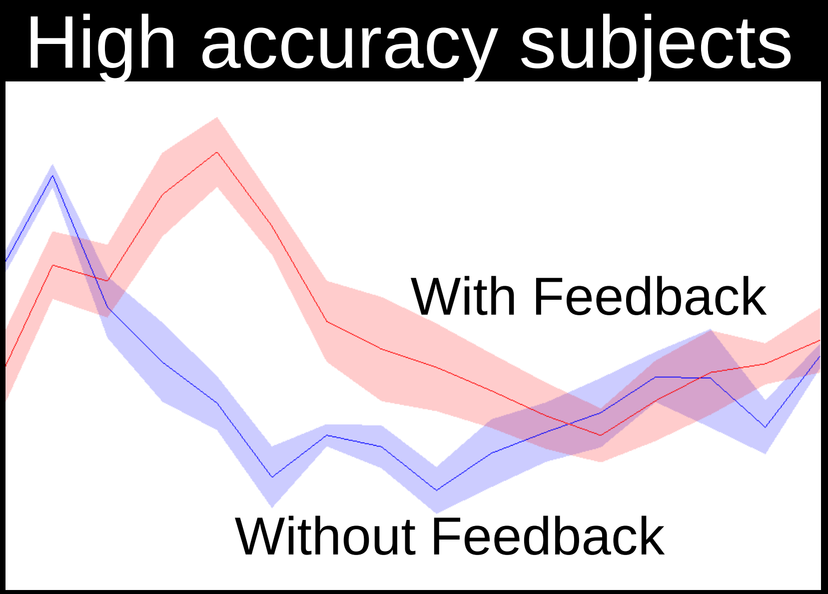
Fast vs. Slow Counting

Crave vs. Don’t Crave

Frontoparietal







Multivoxel pattern-based real-time fMRI

- Conceptual overview
- Experimental flexibility
- Practicalities and additional resources

Computational considerations

First real-time FMRI (Cox et al., 1995), developed a recursive partial correlation algorithm.

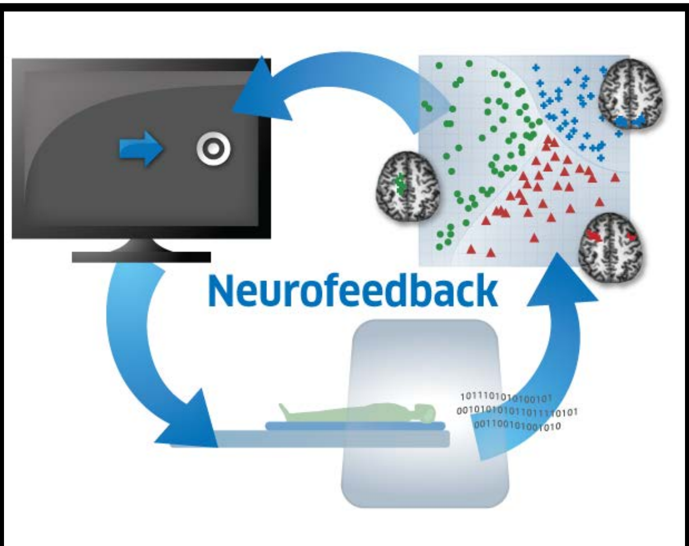
$$\rho = \frac{(\mathbf{Pr})^T(\mathbf{Px})}{|\mathbf{Pr}| \cdot |\mathbf{Px}|} = \frac{\mathbf{r}^T\mathbf{Px}}{(\mathbf{r}^T\mathbf{Pr} \cdot \mathbf{x}^T\mathbf{Px})^{1/2}} \quad \alpha = \frac{\mathbf{r}^T\mathbf{Px}}{\mathbf{r}^T\mathbf{Pr}}. \qquad [2]$$

Equation [2] is not well suited for real-time FMRI…As the number of images grows (i.e., as the vectors increase in dimension), the amount of calculation will grow. In a real-time application, this is unacceptable, because at some point the computer will not be able to finish processing a new image before the next one is ready.

Computational considerations

- Real-time classification
  - Train with a deterministic algorithm
  - Training is computationally intensive but highly doable
    - Converge “immediately” after scan
  - Classify on an image-by-image basis

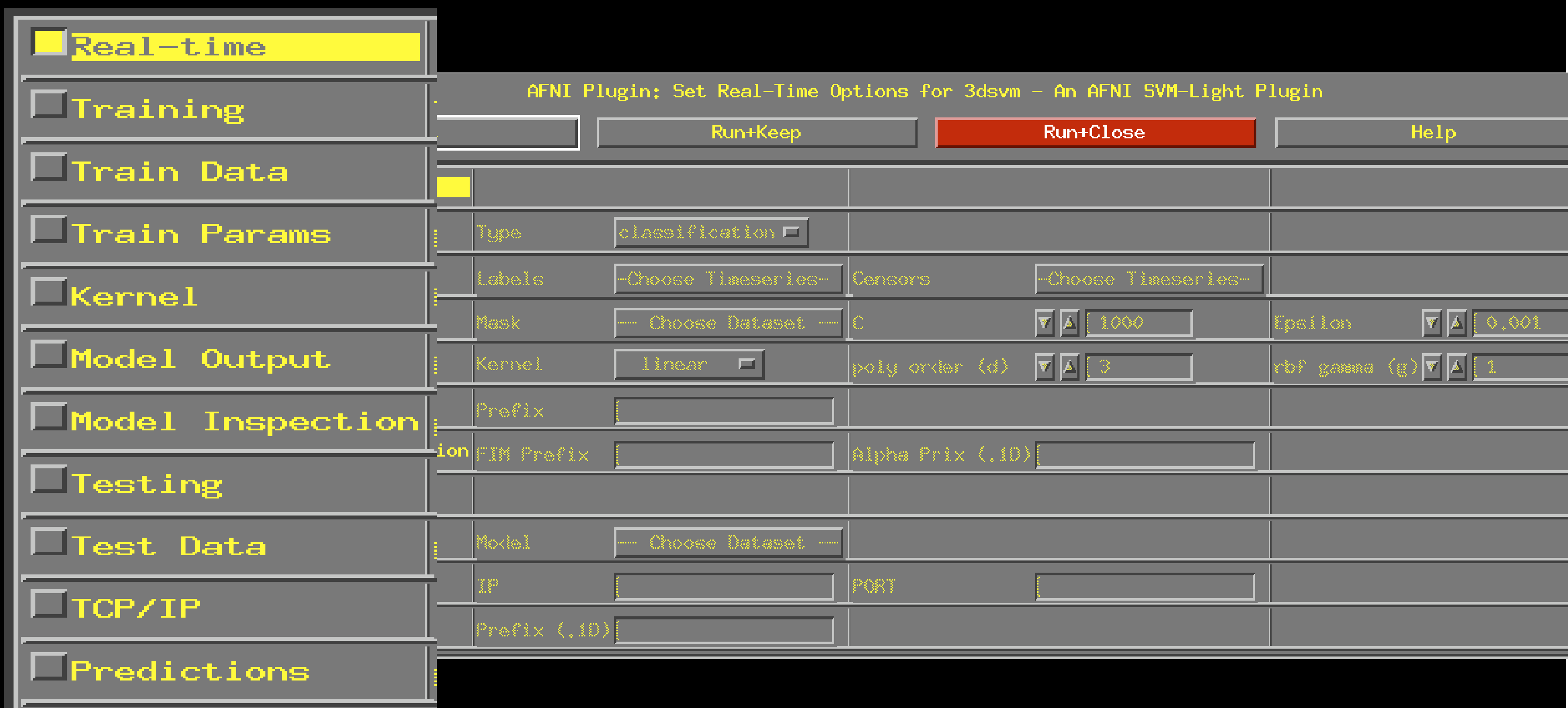
Basic Benchmarks



- Classifier safety factor > 20,000x.
  - Approximately 1µsec / dot product.
- Network/AFNI Transfers > 20x volumes
  - Approximately 100 µsec / slice to transfer

3dsvm Plugin Screenshot

Support Vector Machine Analysis



3dsvm

LaConte Lab

Home | Research | Publications | Facility | People | Links
Slides | 3dsvm

3dsvm - an SVM-Light plugin for AFNI

description | interactive screen shot | download info | data and use
todo | developers | reference

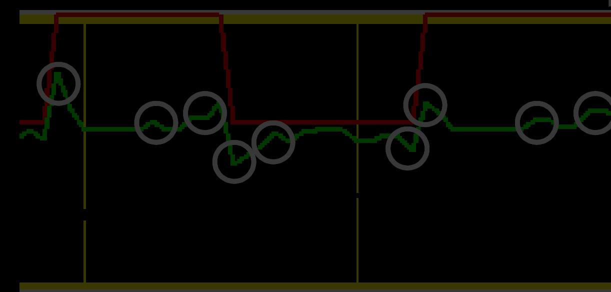
description

3dsvm is a command line program and plugin for AFNI, built around SVM-Light. It provides the ability to analyze functional magnetic resonance imaging (fMRI) data as described in (LaConte et al., 2005).

Features:

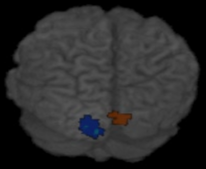
- Reading AFNI-supported binary image formats
- Masking of variables (brain pixels)
- Censoring training samples
- Visualizing alphas as time series and linear weight vectors as functional overlays
- Classifying multiple categories

Supervised learning of fMRI with 3dsvm can be used for predicting brain states to enhance our understanding of brain systems, complementing the conventional emphasis on spatial mapping. The figure below classification formalism used by 3dsvm. For each time point, the brain voxel intensities can be represented in a high-dimensional vector space. During an fMRI experiment, each image is a point in the vector sp of these points, a classification model can be estimated to distinguish between experimental states. After the model is determined, independent data can be assigned a class membership

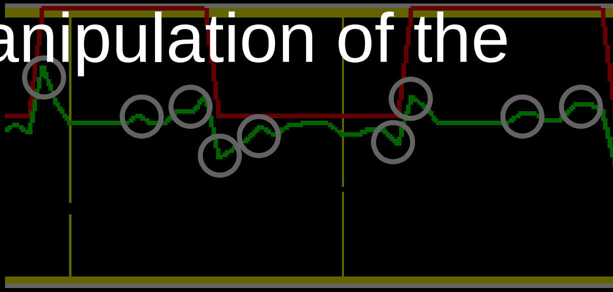


- 3dsvm is a command line program and plugin for AFNI, built around SVM-Light. It provides the ability to analyze functional magnetic resonance imaging (fMRI) data as described in (LaConte et al., 2005)
- lacontelab.org/3dsvm.html

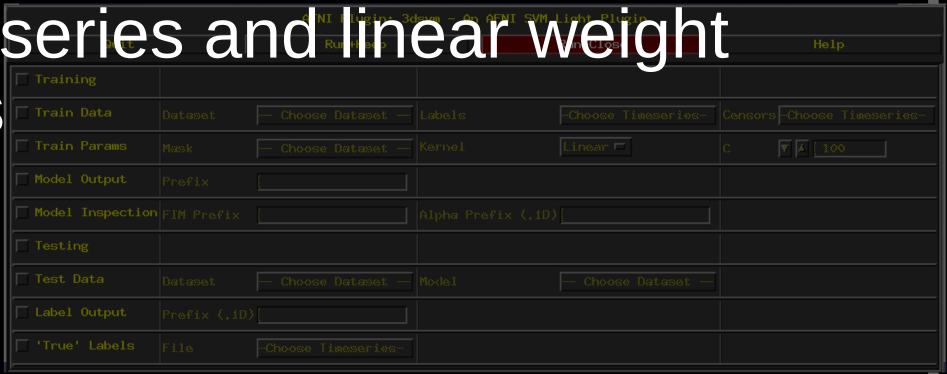
3dsvm features



- Distributed with AFNI
- Reading AFNI-supported formats (including NIfTI)
  - Thus all preprocessing and data manipulation of the major software packages



- Masking of variables (brain pixels)
- Censoring training samples
- Visualizing alphas as time series and linear weight vectors as functional maps



- Multi-class classification
- Regression
- Support for non-linear kernels

3dsvm tour: basic steps

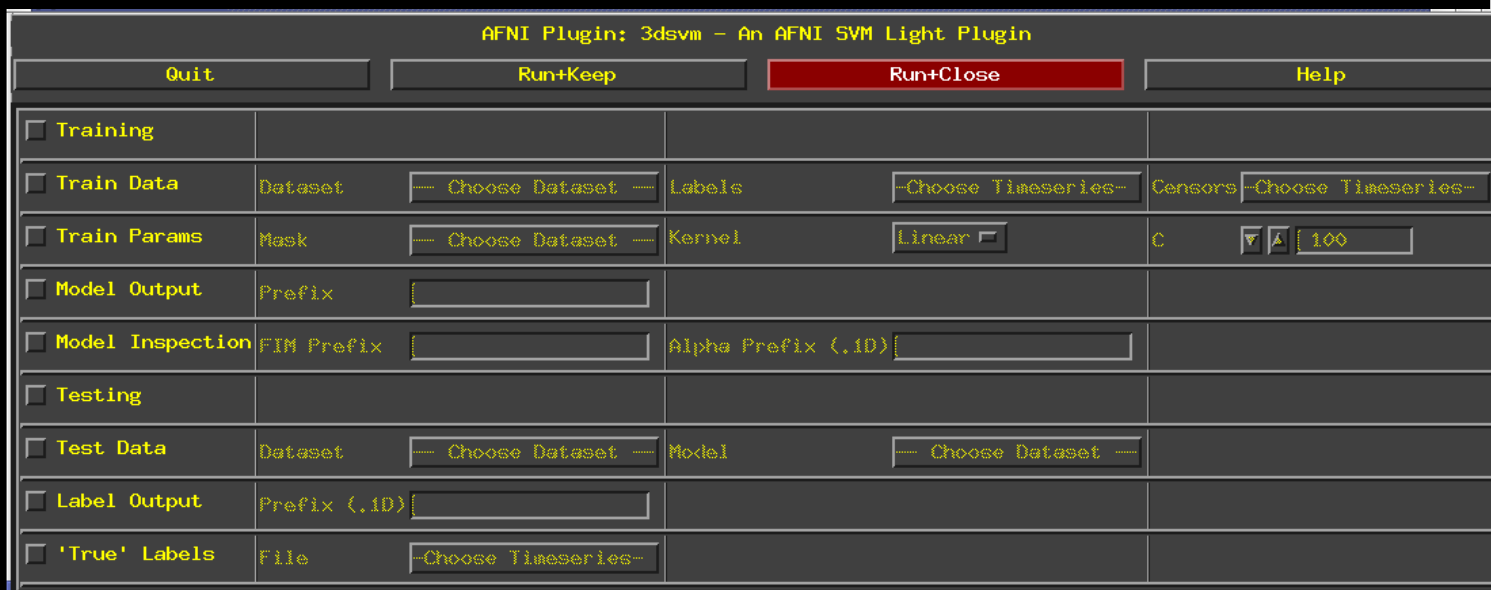
- Prepare training and test data sets
  - fMRI (3D+t)
  - Labels (1D) – labels for test data are optional (needed to to calculate accuracy)
  - Mask for training data (3D) – 3dsvm considers mask to be part of the model it generates
- 3dsvm training
  - Creates a model that can be tested with independent data
  - For convenience, inspecting the model
    - Model alphas (1D)
    - Weight vector map (3D)
- 3dsvm testing
  - Calculates class and/or distance measure for each new timepoint
  - Prediction accuracy (if test set labels are available)

3dsvm Plugin Snapshot

Support Vector Machine Analysis

training

testing



Command Line

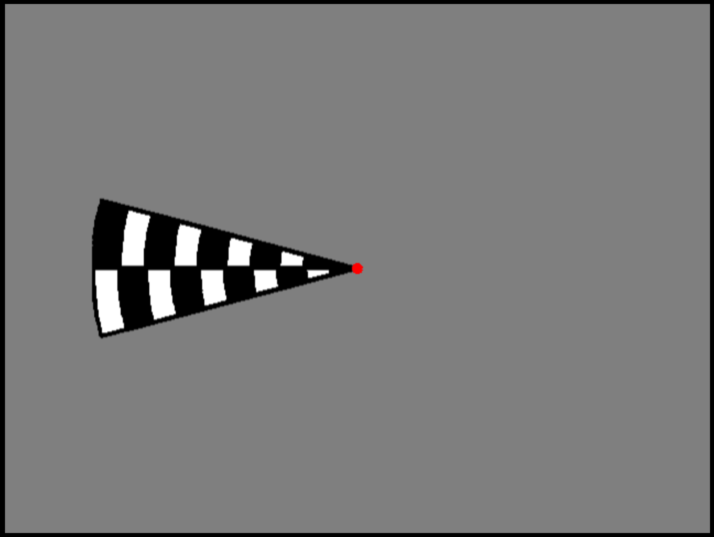
Training - 3dsvm -trainvol run1+orig \
-trainlabels run1_categories.1D \
-mask mask+orig \
-model model_run1

Testing - 3dsvm -testvol run2+orig \
-model model_run1+orig \
-predictions pred2_model1

Example:

Left vs. Right visual stimulus

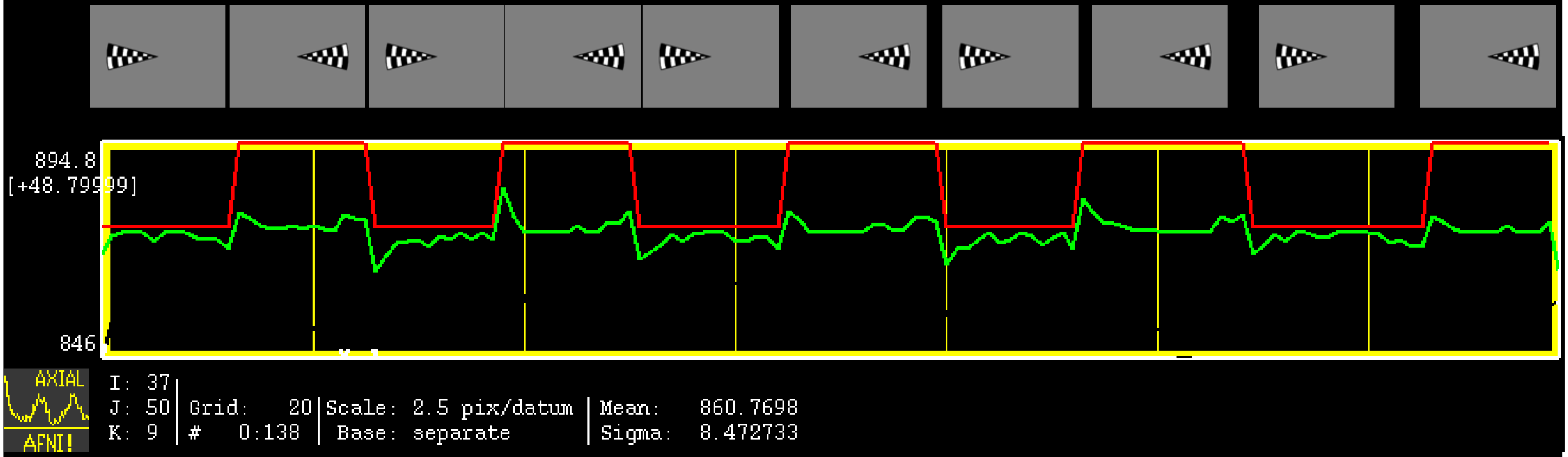
- 3T fmri 31 axial EPI slices, TR/TE = 2000/31 msec, voxel=3.4 X 3.4 X 4 mm).
- Randomized block lengths alternating between left and right stimulus.



Example:

Left vs. Right visual stimulus

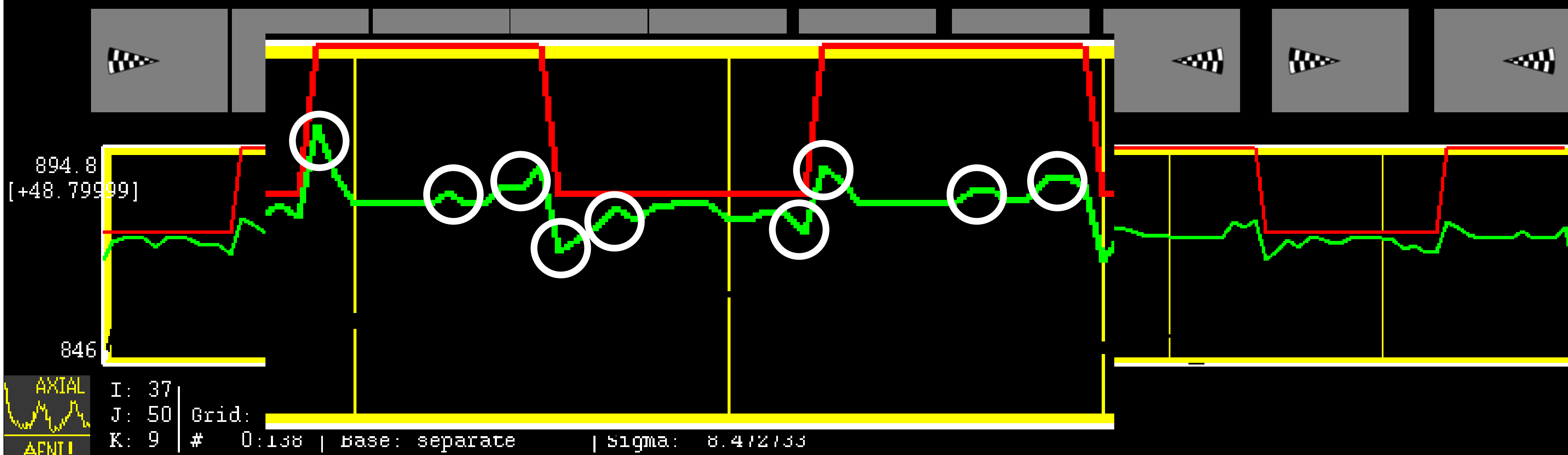
- 3T fmri 31 axial EPI slices, TR/TE = 2000/31 msec, voxel=3.4 X 3.4 X 4 mm).
- Randomized block lengths alternating between left and right stimulus.



Example:

Left vs. Right visual stimulus

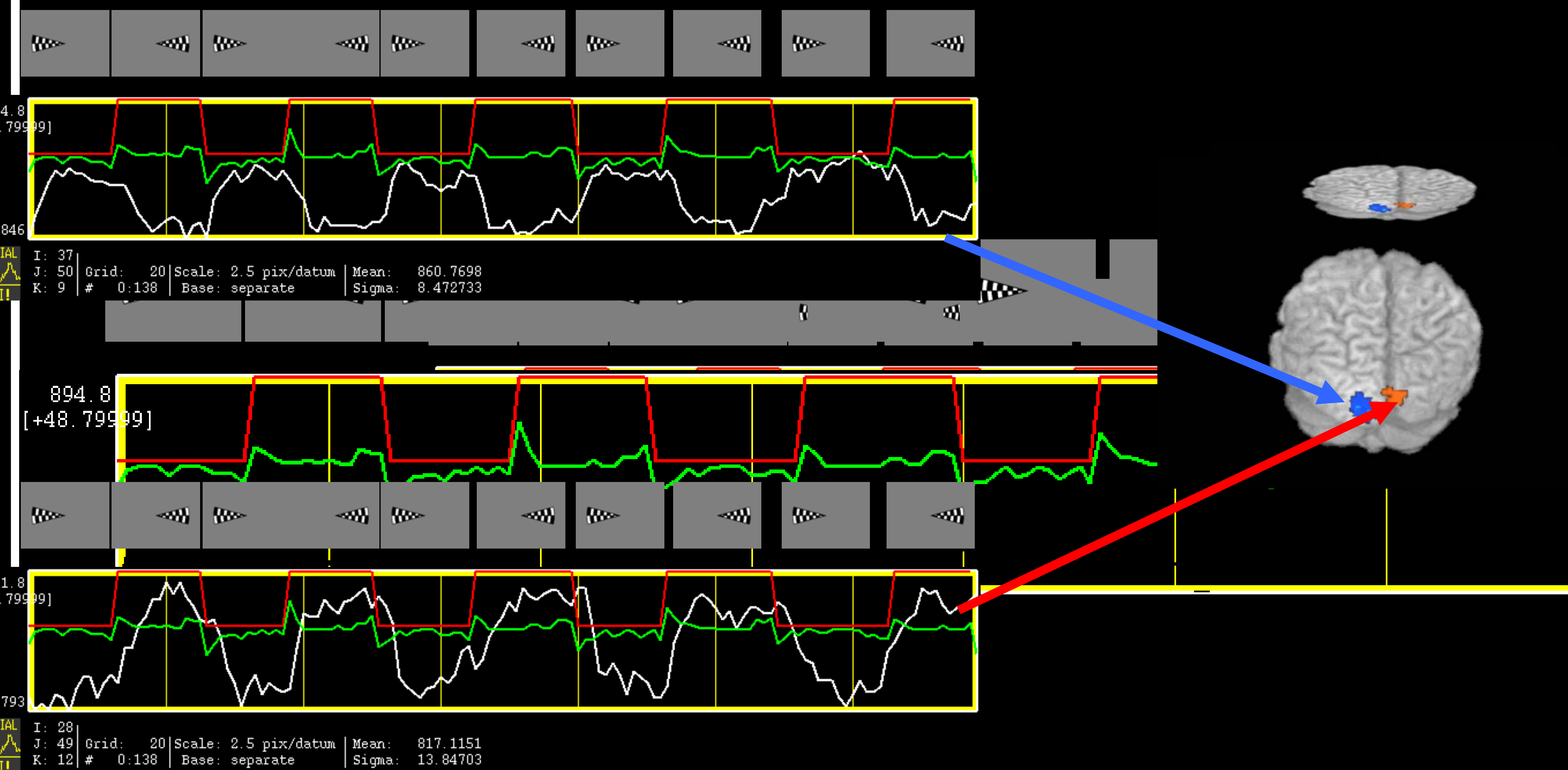
- 3T fmri 31 axial EPI slices, TR/TE = 2000/31 msec, voxel=3.4 X 3.4 X 4 mm).
- Randomized block lengths alternating between left and right stimulus.



Example:

Left vs. Right visual stimulus

- 3T fmri 31 axial EPI slices, TR/TE = 2000/31 msec, voxel=3.4 X 3.4 X 4 mm).
- Randomized block lengths alternating between left and right stimulus.

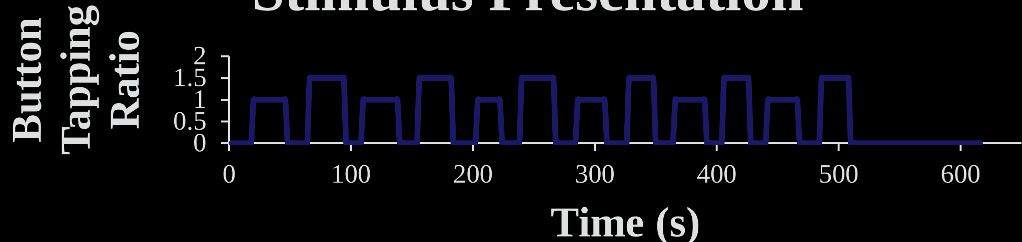


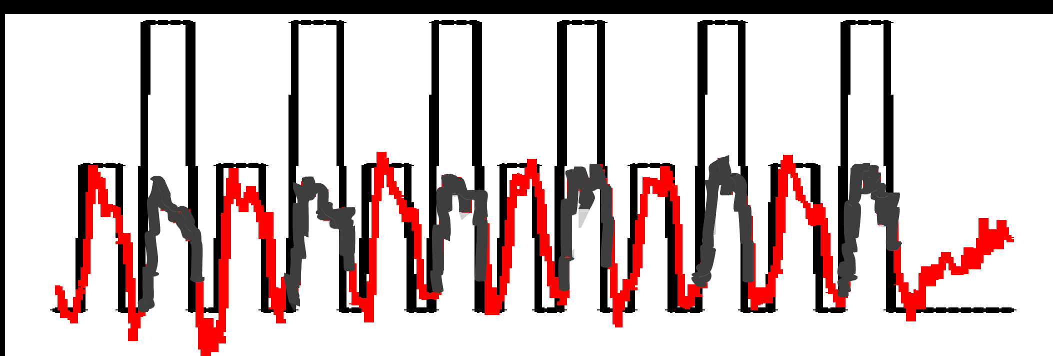
example 2

Bimanual coordination task:

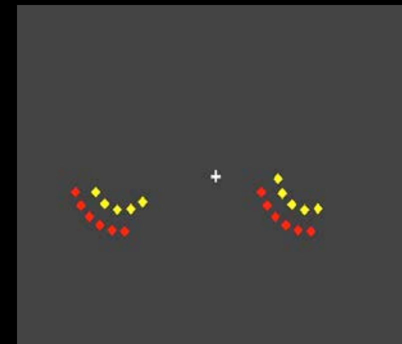
Feedback of both the left and right hand performance was provided by a four arc display described in (Klaiman and Karniel, 2006), where button tapping was used to move the inner arches to match the speed of the outer arches.

Stimulus Presentation

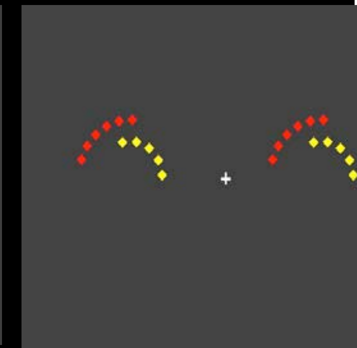




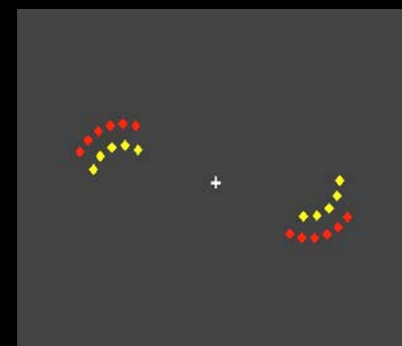
Example Stimulus Frames



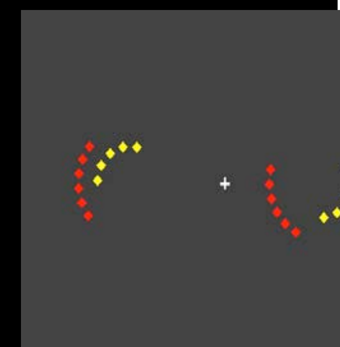
1:1 – Good Performance



1:1 – Poor Performance



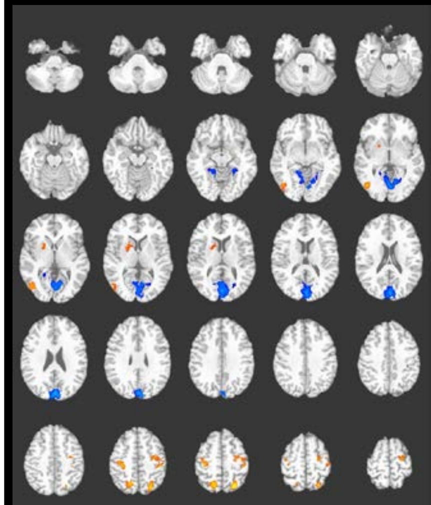
3:2 – Good Performance



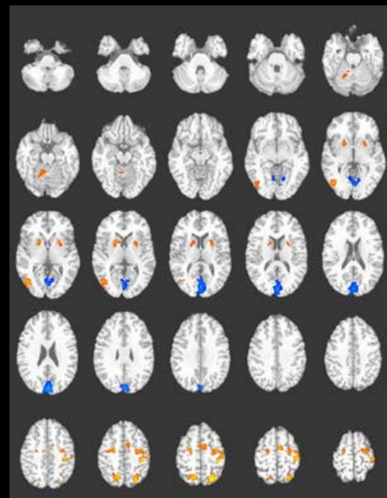
3:2 – Poor Performance

example 2

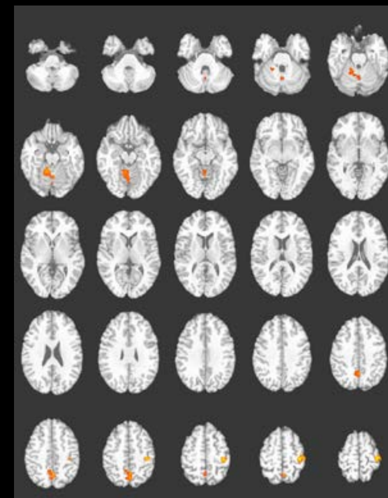
Group brain maps of the SVM models for the 14 subjects



1:1 (red) vs. Fixation (blue)
FDR-corrected; q<0.01



3:2 (red) vs. Fixation (blue)
FDR-corrected; q<0.01



3:2 (red) vs. 1:1 (blue)
p=0.001

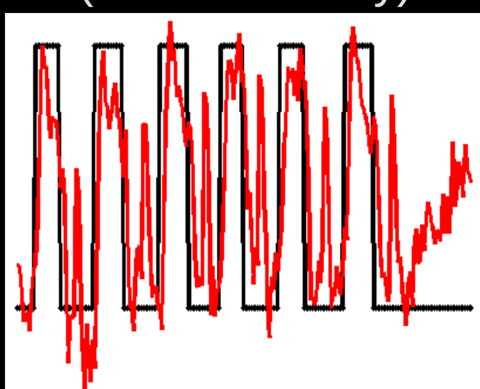
Distributed pattern for 1:1 & 3:2 rhythms:
- Bil. Precuneus (BA7)
- L. (1:1) & Bil. (3:2) postcentral (BA4)
- Bil. Lentiform
- R. culmen

Distributed pattern during Fixation:
- L. cuneus (BA18)

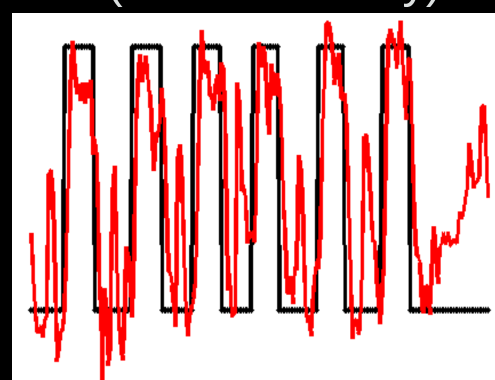
Distributed pattern unique to 1:1 rhythm:
- Bil. Precentral/M1 (BA4)
- R. inf. Temporal (BA37)
- L. sup. Parietal

Distributed pattern unique to 3:2 rhythm:
- R. middle frontal (BA6)
- L. medial frontal (BA6)

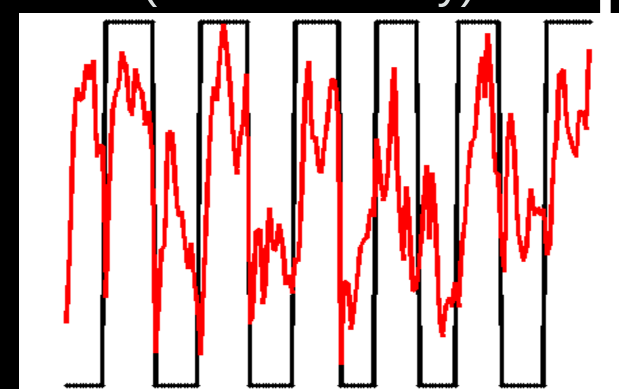
1:1 vs. Fixation
(93% Accuracy)



3:2 vs. Fixation
(93% Accuracy)



3:2 vs. 1:1
(77% Accuracy)

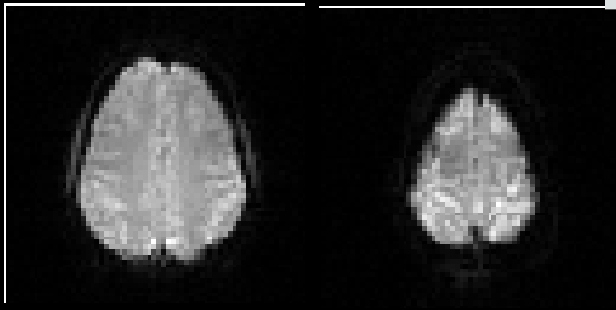


Raw classifier output plotted over label files for a relatively accurate subject. The first two TRs have been removed for each transition between stimuli.

example 2

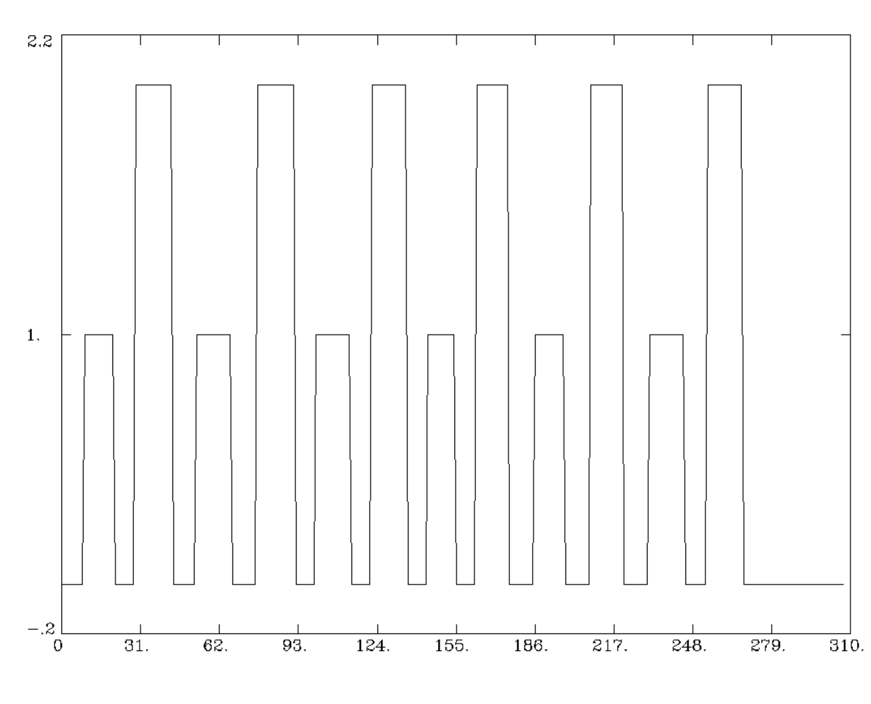
```
#!/bin/csh
# example by Prashant Prasad

3dsvm -trainvol volreg_run1_PPA+orig \
        -trainlabels LABEL_PPA_1.1D \
        -mask automask_run1_PPA+orig \
        -bucket bucket_run1_PPA \
        -model model_run1_PPA

3dsvm -classout \
         -testvol volreg_run2_PPA+orig \
         -testlabels LABEL_PPA_2.1D \
         -model model_run1_PPA+orig \
         -predictions pred_run2_frmRun1_classout

3dsvm -testvol volreg_run2_PPA+orig \
        -testlabels LABEL_PPA_2.1D \
        -model model_run1_PPA+orig \
        -predictions pred_run2_frmRun1
```

example 2



```
#!/bin/csh
# example by Prashant Prasad

3dsvm -trainvol volreg_run1_PPA+orig \
         -trainlabels LABEL_PPA_1.1D \
         -mask automask_run1_PPA+orig \
         -bucket bucket_run1_PPA \
         -model model_run1_PPA

3dsvm -classout \
          -testvol volreg_run2_PPA+orig \
          -testlabels LABEL_PPA_2.1D \
          -model model_run1_PPA+orig \
          -predictions pred_run2_frmRun1_classout

3dsvm -testvol volreg_run2_PPA+orig \
         -testlabels LABEL_PPA_2.1D \
         -model model_run1_PPA+orig \
         -predictions pred_run2_frmRun1
```

```
#!/bin/csh
# example by Prashant Prasad

3dsvm -trainvol volreg_run1_PPA+orig \
          -trainlabels LABEL_PPA_1.1D \
          -mask automask_run1_PPA+orig \
          -bucket bucket_run1_PPA \
          -model model_run1_PPA

3dsvm -classout \
           -testvol volreg_run2_PPA+orig \
           -testlabels LABEL_PPA_2.1D \
           -model model_run1_PPA+orig \
           -predictions pred_run2_frmRun1_classout

3dsvm -testvol volreg_run2_PPA+orig \
          -testlabels LABEL_PPA_2.1D \
          -model model_run1_PPA+orig \
          -predictions pred_run2_frmRun1
```



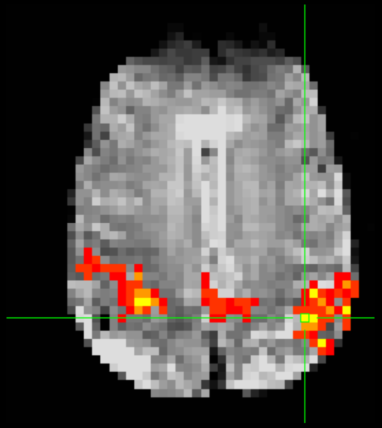

```
0
0
0
0
0
0
0
0
0
1
1
1
1
1
1
1
1
1
1
1
1
0
0
0
0
0
0
0
0
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
0
0
0
0
```

example 2

```
#!/bin/csh
# example by Prashant Prasad

3dsvm -trainvol volreg_run1_PPA+orig \
      -trainlabels LABEL_PPA_1.1D \
      -mask automask_run1_PPA+orig \
      -bucket bucket_run1_PPA \
      -model model_run1_PPA

3dsvm -classout \
       -testvol volreg_run2_PPA+orig \
       -testlabels LABEL_PPA_2.1D \
       -model model_run1_PPA+orig \
       -predictions pred_run2_frmRun1_classout

3dsvm -testvol volreg_run2_PPA+orig \
      -testlabels LABEL_PPA_2.1D \
      -model model_run1_PPA+orig \
      -predictions pred_run2_frmRun1
```

example 2

```
#!/bin/csh
# example by Prashant Prasad

3dsvm -trainvol volreg_run1_PPA+orig \
          -trainlabels LABEL_PPA_1.1D \
          -mask automask_run1_PPA+orig \
          -bucket bucket_run1_PPA \
          -model model_run1_PPA

3dsvm -classout \
           -testvol volreg_run2_PPA+orig \
           -testlabels LABEL_PPA_2.1D \
           -model model_run1_PPA+orig \
           -predictions pred_run2_frmRun1_classout

3dsvm -testvol volreg_run2_PPA+orig \
          -testlabels LABEL_PPA_2.1D \
          -model model_run1_PPA+orig \
          -predictions pred_run2_frmRun1
```



example 2

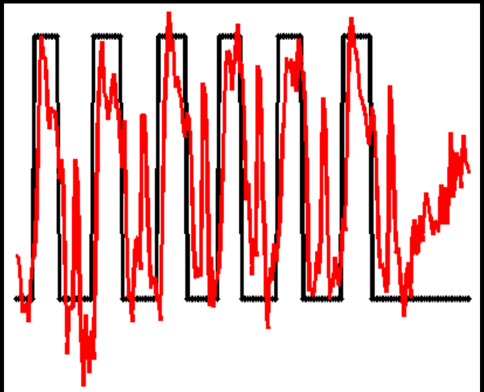
```
#!/bin/csh
# example by Prashant Prasad

3dsvm -trainvol volreg_run1_PPA+orig \
          -trainlabels LABEL_PPA_1.1D \
          -mask automask_run1_PPA+orig \
          -bucket bucket_run1_PPA \
          -model model_run1_PPA

3dsvm -classout \
           -testvol volreg_run2_PPA+orig \
           -testlabels LABEL_PPA_2.1D \
           -model model_run1_PPA+orig \
           -predictions pred_run2_frmRun1_classout

3dsvm -testvol volreg_run2_PPA+orig \
          -testlabels LABEL_PPA_2.1D \
          -model model_run1_PPA+orig \
          -predictions pred_run2_frmRun1
```

example 2

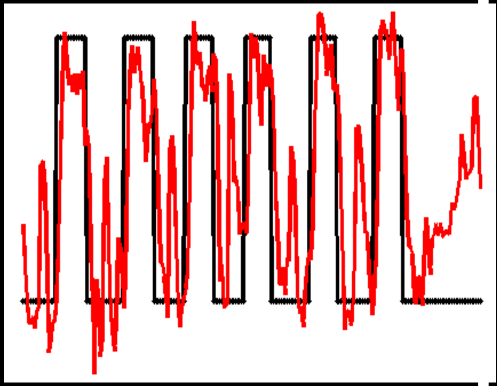
```
#!/bin/csh
# example by Prashant Prasad

3dsvm -trainvol volreg_run1_PPA+orig \
          -trainlabels LABEL_PPA_1.1D \
          -mask automask_run1_PPA+orig \
          -bucket bucket_run1_PPA \
          -model model_run1_PPA

3dsvm -classout \
           -testvol volreg_run2_PPA+orig \
           -testlabels LABEL_PPA_2.1D \
           -model model_run1_PPA+orig \
           -predictions pred_run2_frmRun1_classout

3dsvm -testvol volreg_run2_PPA+orig \
          -testlabels LABEL_PPA_2.1D \
          -model model_run1_PPA+orig \
          -predictions pred_run2_frmRun1
```

```
#!/bin/csh
# example by Prashant Prasad

3dsvm -trainvol volreg_run1_PPA+orig \
        -trainlabels LABEL_PPA_1.1D \
        -mask automask_run1_PPA+orig \
        -bucket bucket_run1_PPA \
        -model model_run1_PPA

3dsvm -classout \
         -testvol volreg_run2_PPA+orig \
         -testlabels LABEL_PPA_2.1D \
         -model model_run1_PPA+orig \
         -predictions pred_run2_frmRun1_classout

3dsvm -testvol volreg_run2_PPA+orig \
        -testlabels LABEL_PPA_2.1D \
        -model model_run1_PPA+orig \
        -predictions pred_run2_frmRun1
```

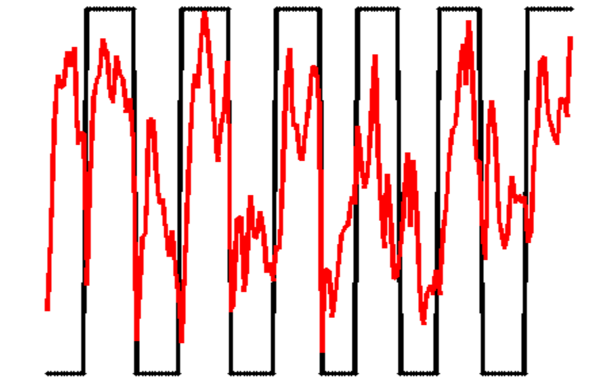
1:1 vs. Fixation
(93% Accuracy)



3:2 vs. Fixation
(93% Accuracy)



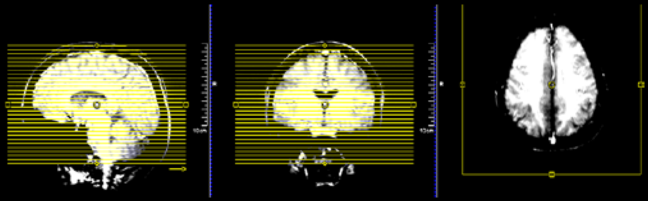
3:2 vs. 1:1
(77% Accuracy)



Demonstration Experiment

Initial scans

- **Localizer** (9 seconds)

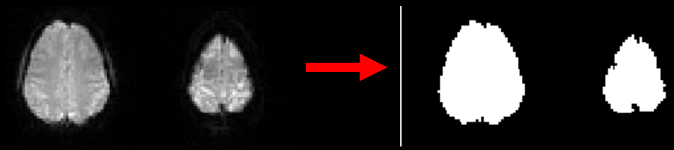


- **Anatomical scan** (4.5 minutes)

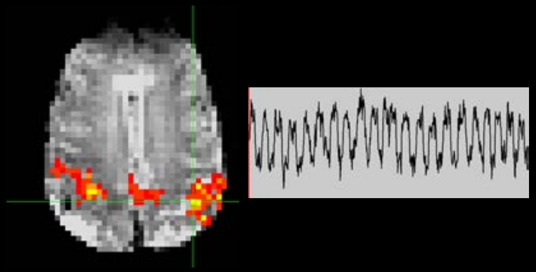
LaConte, *NeuroImage* (2011)

fMRI runs

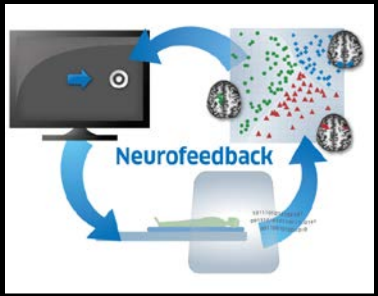
- **Masking run** (10 seconds)



- **Training run** (6 minutes)

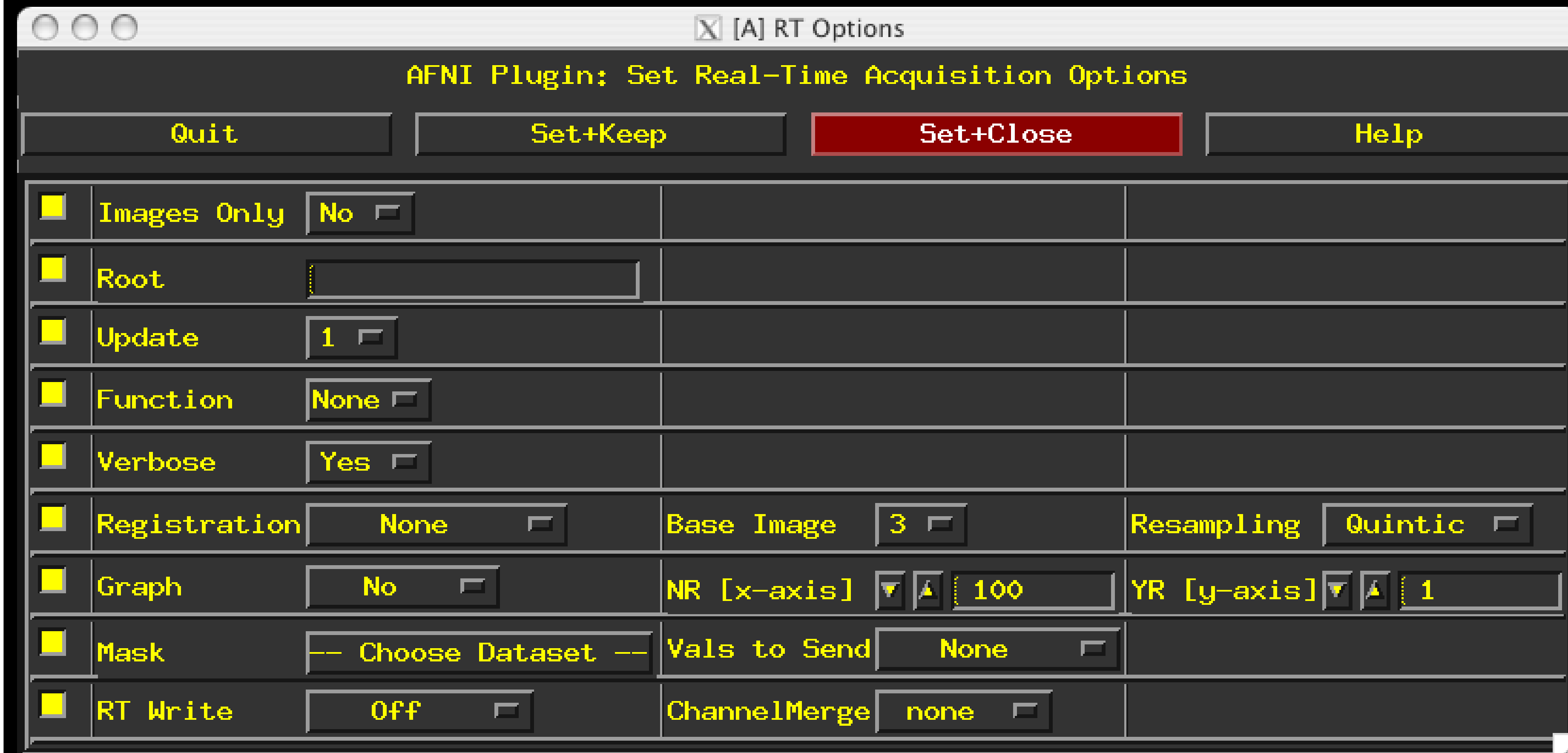


- **Feedback run** (6 minutes)



Setting up AFNI's RT plugin

- Manually
  - Good for learning and demo



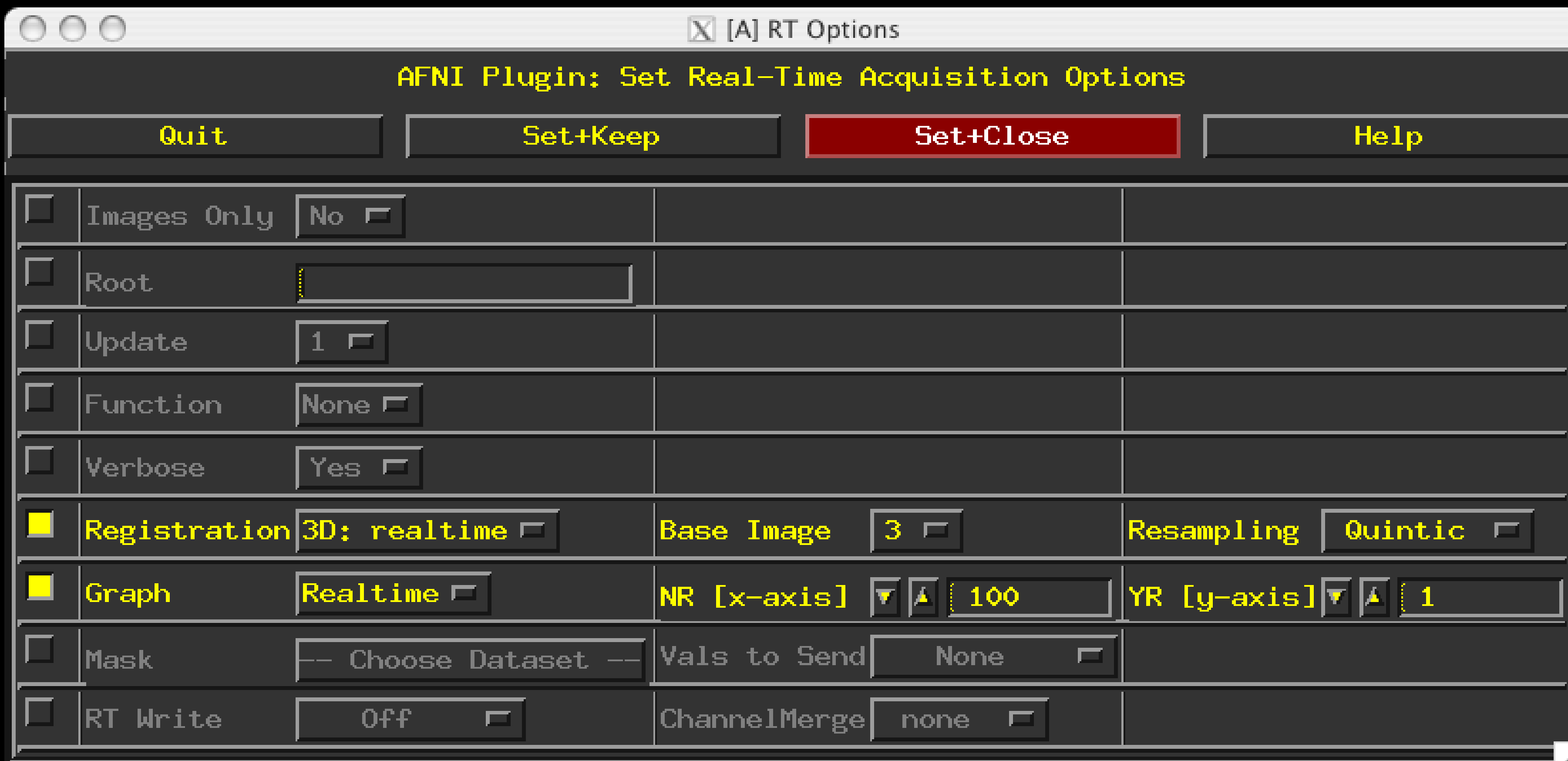
Slide provided by Ziad Saad

Setting up AFNI's RT plugin

- Via Environment Variables

  setenv AFNI_REALTIME_Registration 3D:_realtime

  setenv AFNI_REALTIME_Graph Realtime



Slide provided by Ziad Saad

AFNI help resources

- Readme files
  - README.driver
  - README.environment
  - README.realtime
- Demo material available on:

  http://afni.nimh.nih.gov
- Automation
  - @DriveAfni script
  - @DriveSuma script
  - @DO.examples
- Sample programs
  - rtfeedme.c
  - Dimon.c
  - serial_helper.c
  - realtime_receiver.py

Slide provided by Ziad Saad

Multivoxel pattern-based real-time fMRI

- Conceptual overview
  - Supervised learning for fMRI
    - (classification and regression)
  - Integration with MRI and real-time platforms
- Experimental flexibility
  - Whole-brain models trained to the distributed patterns of each individual for neurofeedback, BCIs, adaptive fMRI paradigms, etc.
  - Feedback improves classifications
    - Employs frontoparietal networks
- Practicalities and additional resources
  - 3dsvm and AFNI’s realtime plugin