Spatial Perspective Taking for Developmental Robotics

Yihan Zhang
7/26/2017

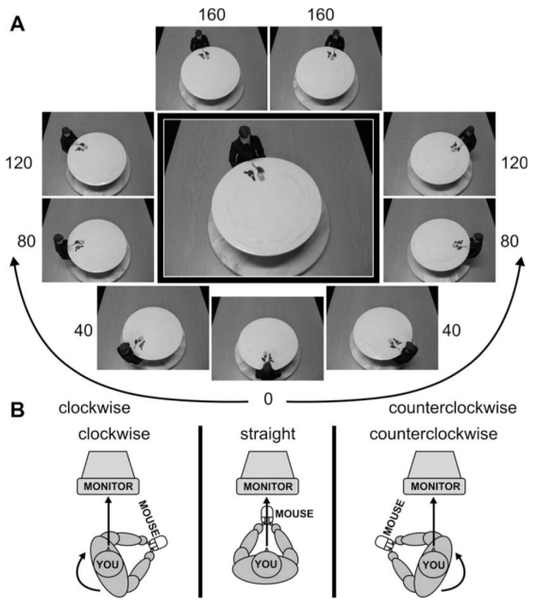
Agenda

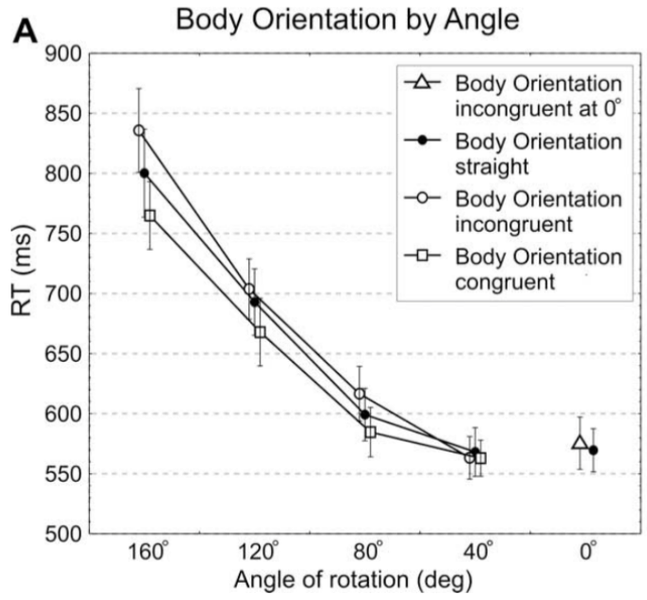
- Background Study
- Hypothesis (original and modified)
- Experiment Setting
- Model Design
- Result Analysis
- Interesting Stuff (*)
    - Phenomenons that won’t affect the purpose of this study, but interesting to know

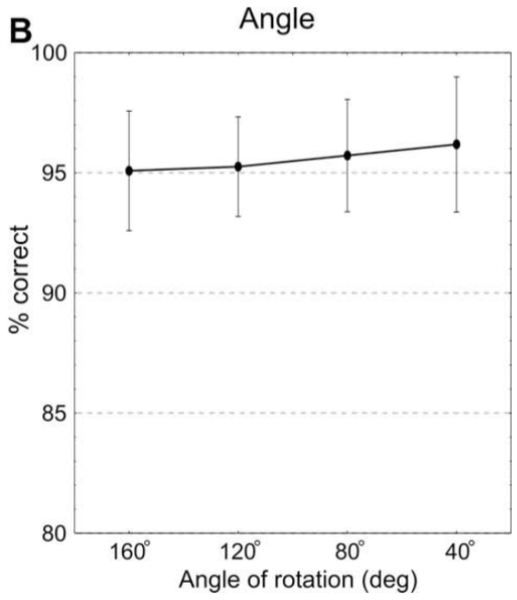
The embodied nature of spatial perspective taking: Embodied transformation versus sensorimotor interference

[Kessler & Thomson, 2010]

- Experiment Design & Result Analysis (Experiment 1)







- At lower angles: visual matching. At higher angles: mental self-rotation.
- No embodiment effect was observed at 40°.
- Overall conclusion: When doing SPT, human endogenously initiate self-rotation. Exogenous embodiment process at the end of self-rotation to align the posture.

Hypothesis (original)

- When angular discrepancy is low, body schema contains enough visual information to do spatial perspective taking
    - Low angular discrepancy visual stream, such as 45° and 315°, are naturally similar to own sensorimotor experience
- When angular discrepancy is high, mental self-rotation is required
    - A common goal could be the coordinate for such transformation
- Following experiments conducted do not show evidence supporting this idea

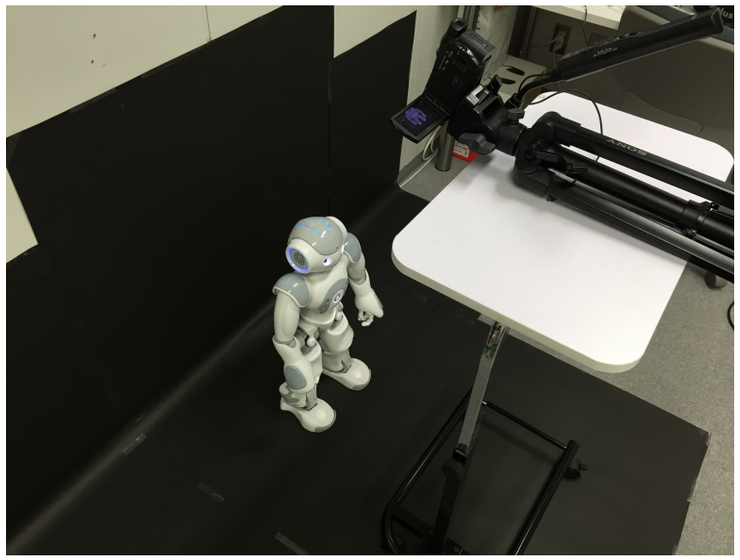
Experiment Setting

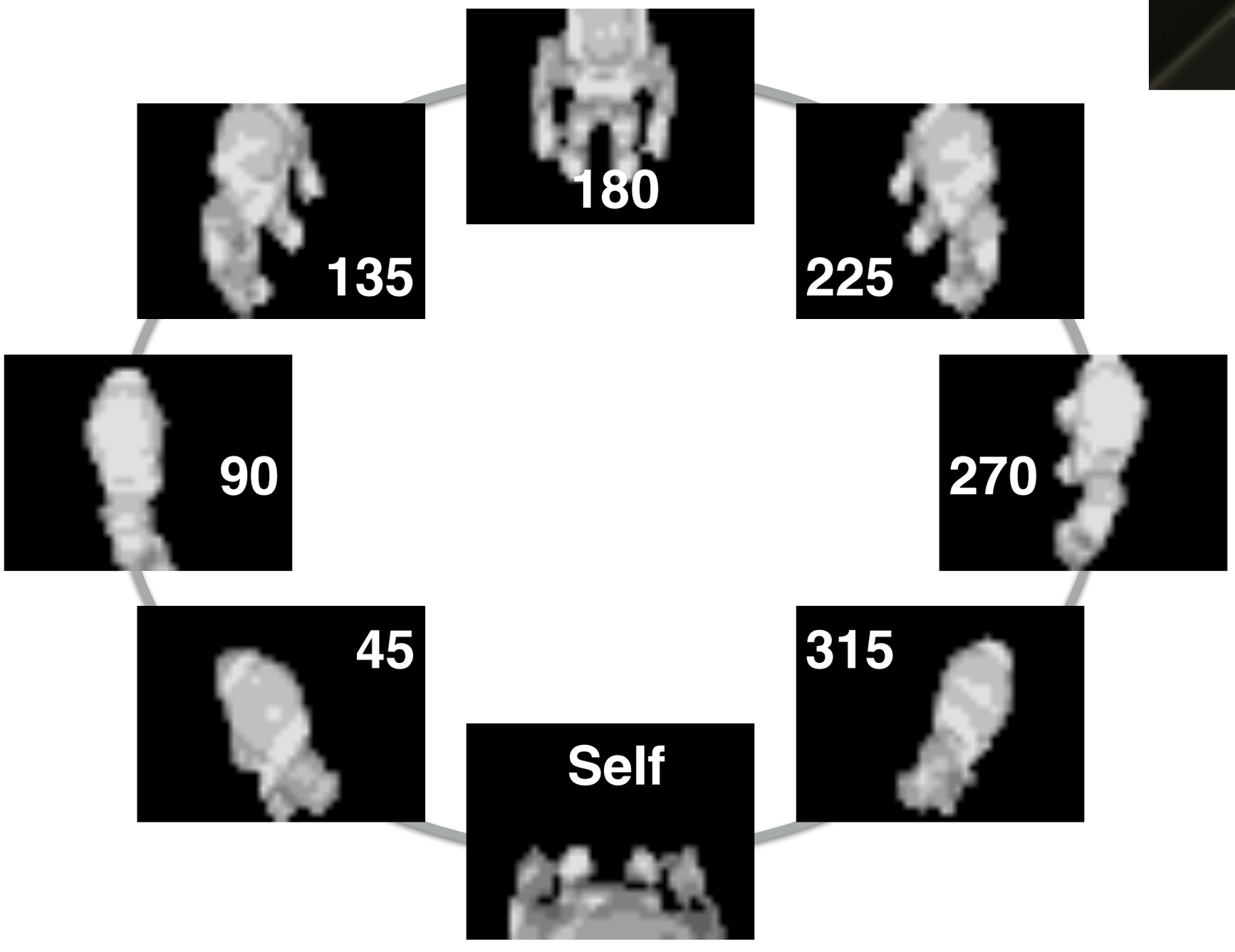
- 3 motion sequences designed
  - Data set 1 (both hands raise):
    - raise right hand for two times
    - left hand for two times
    - both hands for two times
  - Data set 2 (left hand raise):
    - raise left hand for four times
  - Data set 3 (right hand raise):
    - raise right hand for four times

- Data set 1:

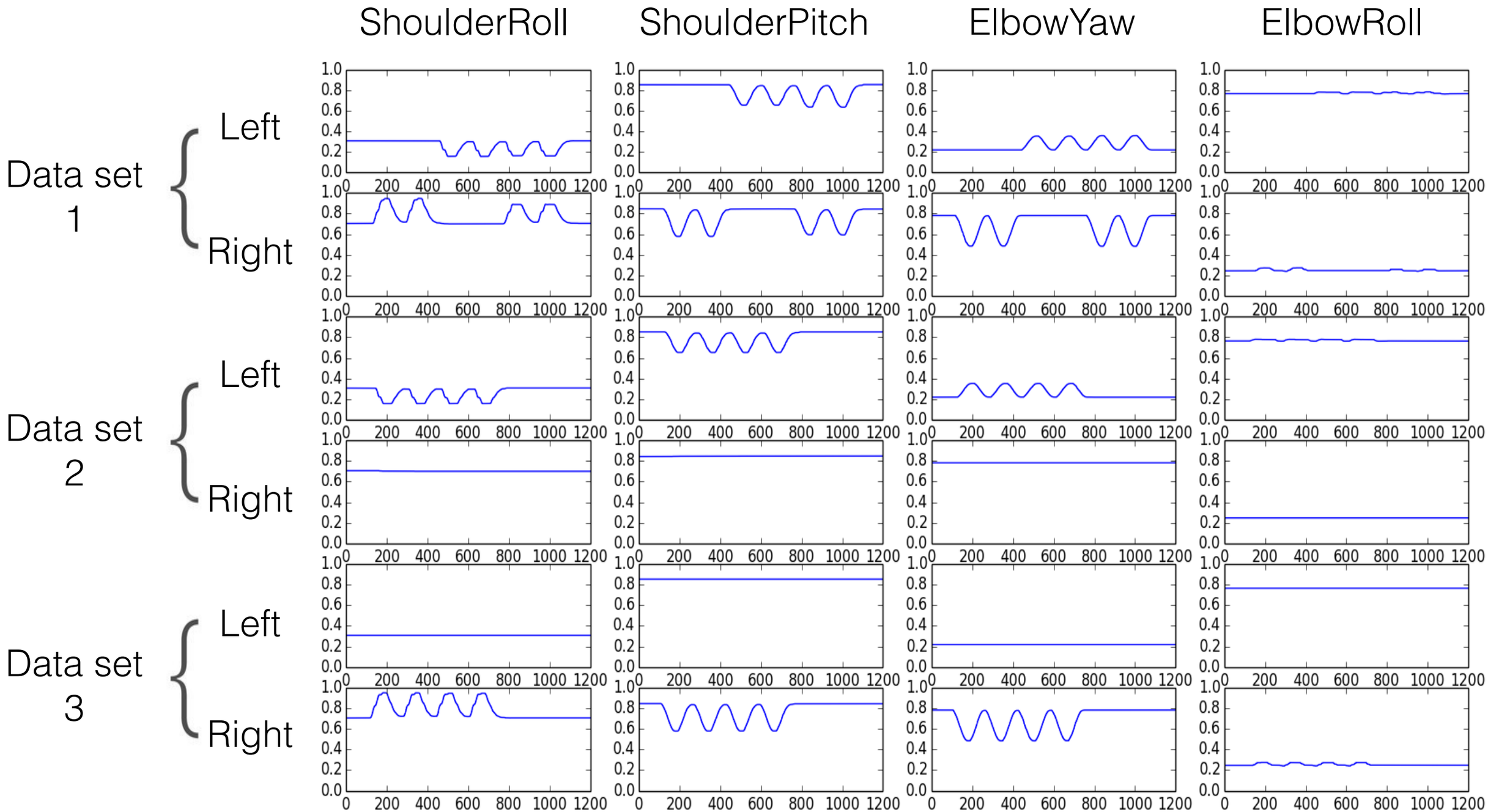
  - fps: 100
  - size: 40 * 30
  - color: 8 bits grayscale
  - timesteps: 1200

- 8 different perspectives recorded
  - own perspective
  - every 45 degrees clockwise

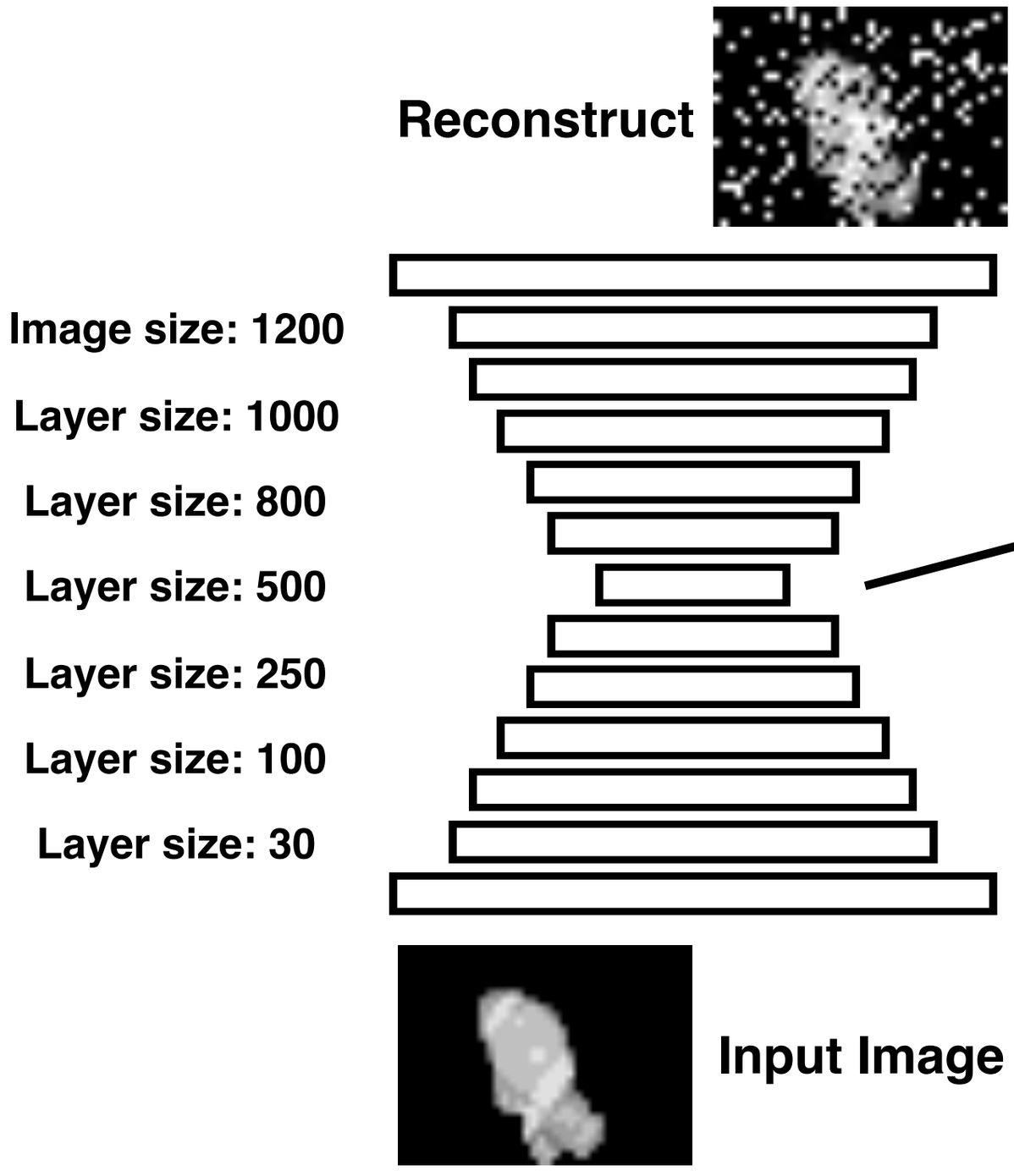




Joint Data


ShoulderRoll
ShoulderPitch
ElbowYaw
ElbowRoll
Data set 1
Left
Right
Data set 2
Left
Right
Data set 3
Left
Right

Visual Data Compression Using Autoencoder



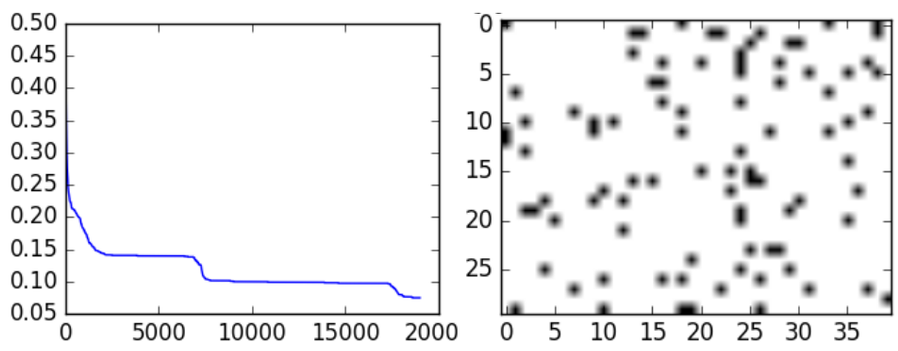
30 dimensional encode, compressed from 1200 dimensional image

Optional Upgrade: (*) Denoising Autoencoder — Train the dA to predict correct input from randomly corrupted ones

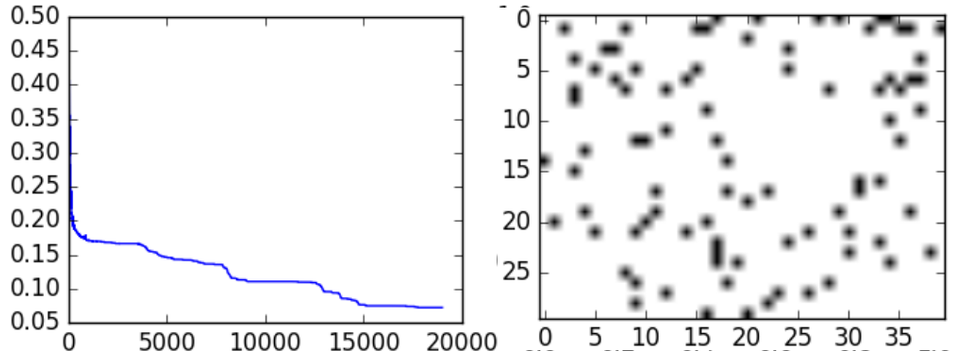
Data Compression Using Autoencoder (con't)

- Original visual stream vs Reconstructed visual stream

- Potential advantage of Denoising Autoencoder (*)
  - easier to escape from local optima because of random noise. (toy example of reconstructing all white image)



Autoencoder



Denoising Autoencoder

Evaluation for Data Processing

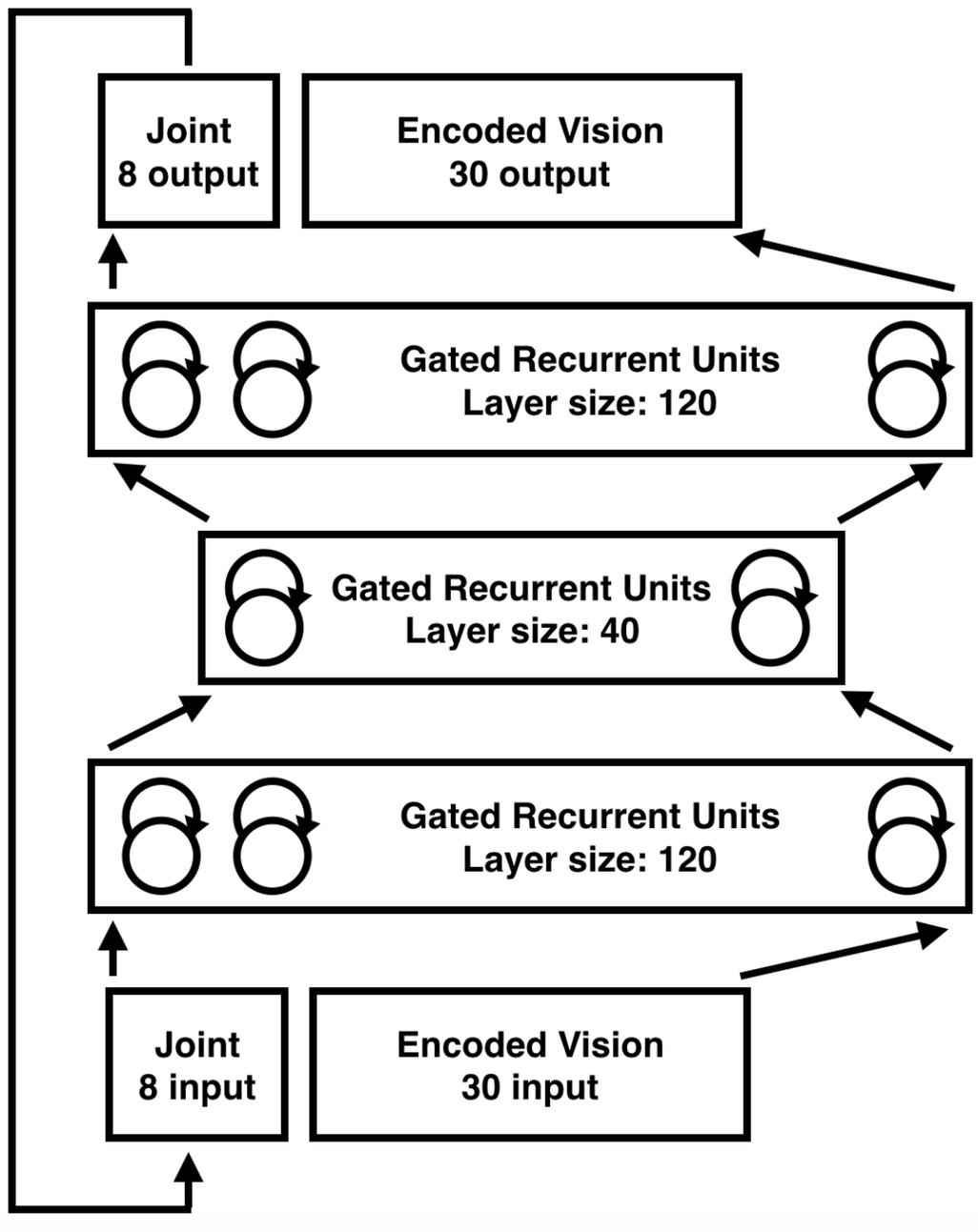
- There is a static noise among every reconstructed data
  - local optima produced by autoencoder
  - maybe caused by inappropriate loss function (mean squared error)
- However, since it’s static, RNN won’t pay much attention on the noise (in theory)

Note: GRU are not nodes, this graph is only for conceptual understanding
The GRU layer computed in the matrix manner

Model Design

- Input to the model:
  - V(t): visual data input stream
  - J(t): joint predicted from (t-1)
- Output of the model:
  - V(t+1): predicted visual data
  - J(t+1): predicted joint data
- Loss function:
  - Mean squared error
  - For self: both J and V
  - For non-self: only V

- Implemented using Tensorflow

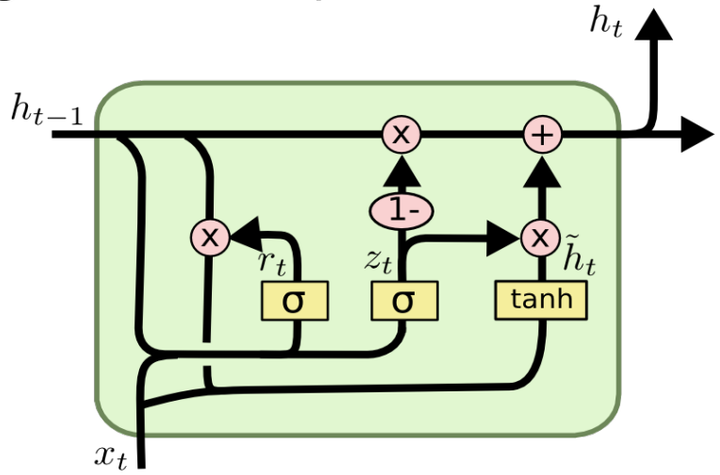


Model Design (con't)

- During training phase
  - Joint reset to starting position
  - Input visual data stream
  - BPTT the correct visual data in next time step
    - If there is a joint feedback, BPTT the correct joint data
- During testing phase
  - Joint reset to starting position
  - Input visual data stream
  - Collect the joint prediction from the closed loop

What is GRU? (*)

- A simpler version of LSTM, which has a better ability at learning long term dependencies than vanilla RNN



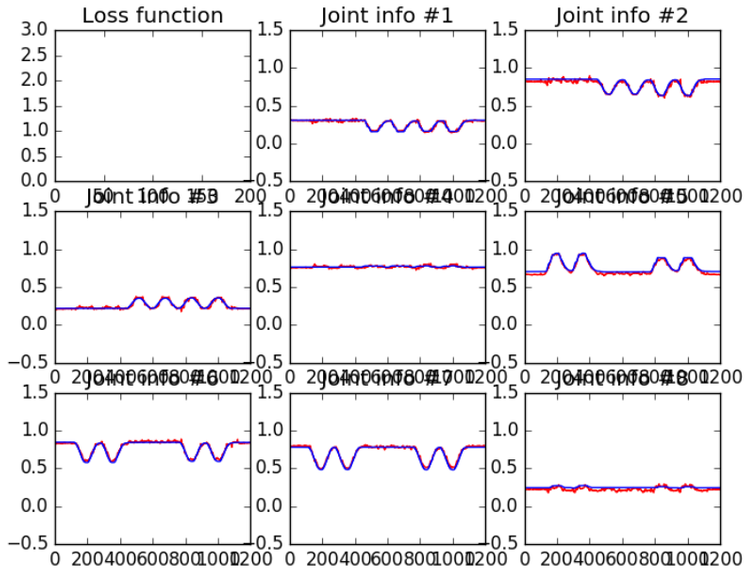
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

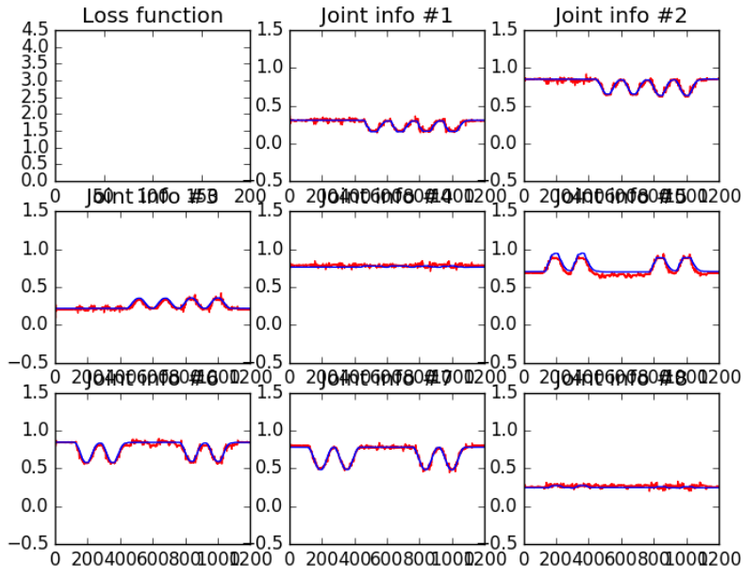
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

- Slightly better performance than RNN in same conditions



GRU

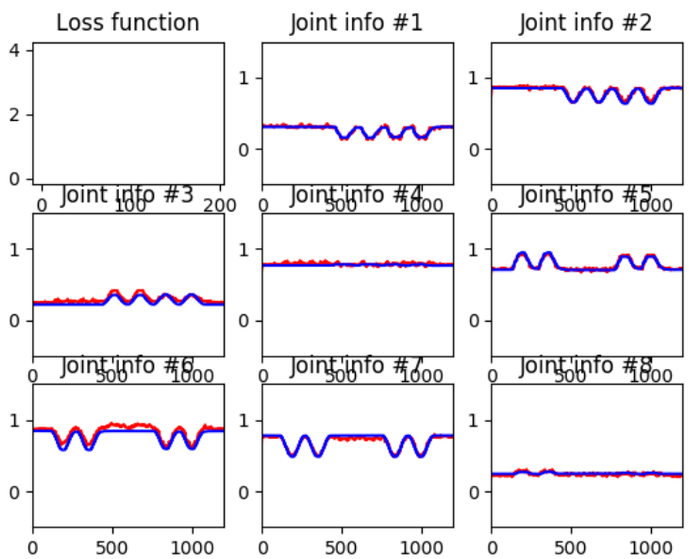


RNN

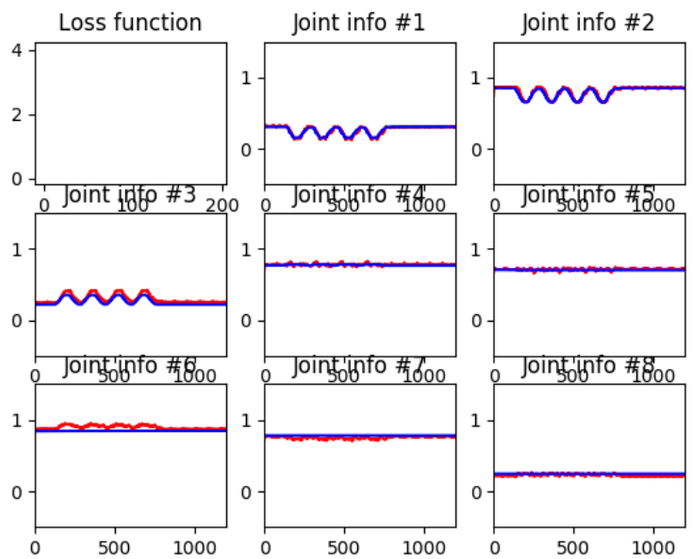
Note: Blue line shows the actual joint data
red line shows predicted joint data from visual input stream

Does this model work?

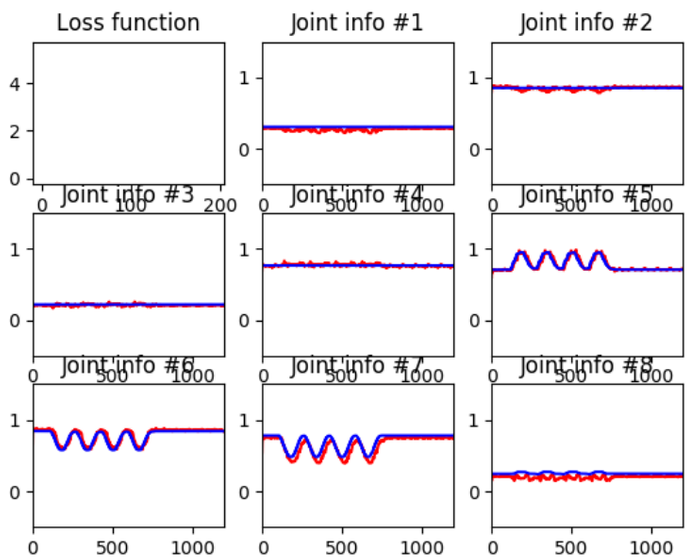
- Toy example:
  - Train the model with data set 1 (both hand raise)
  - Test the model with data set 2 & 3 (left and right hand raise)



- Data set 1(both)



- Data set 2(left)

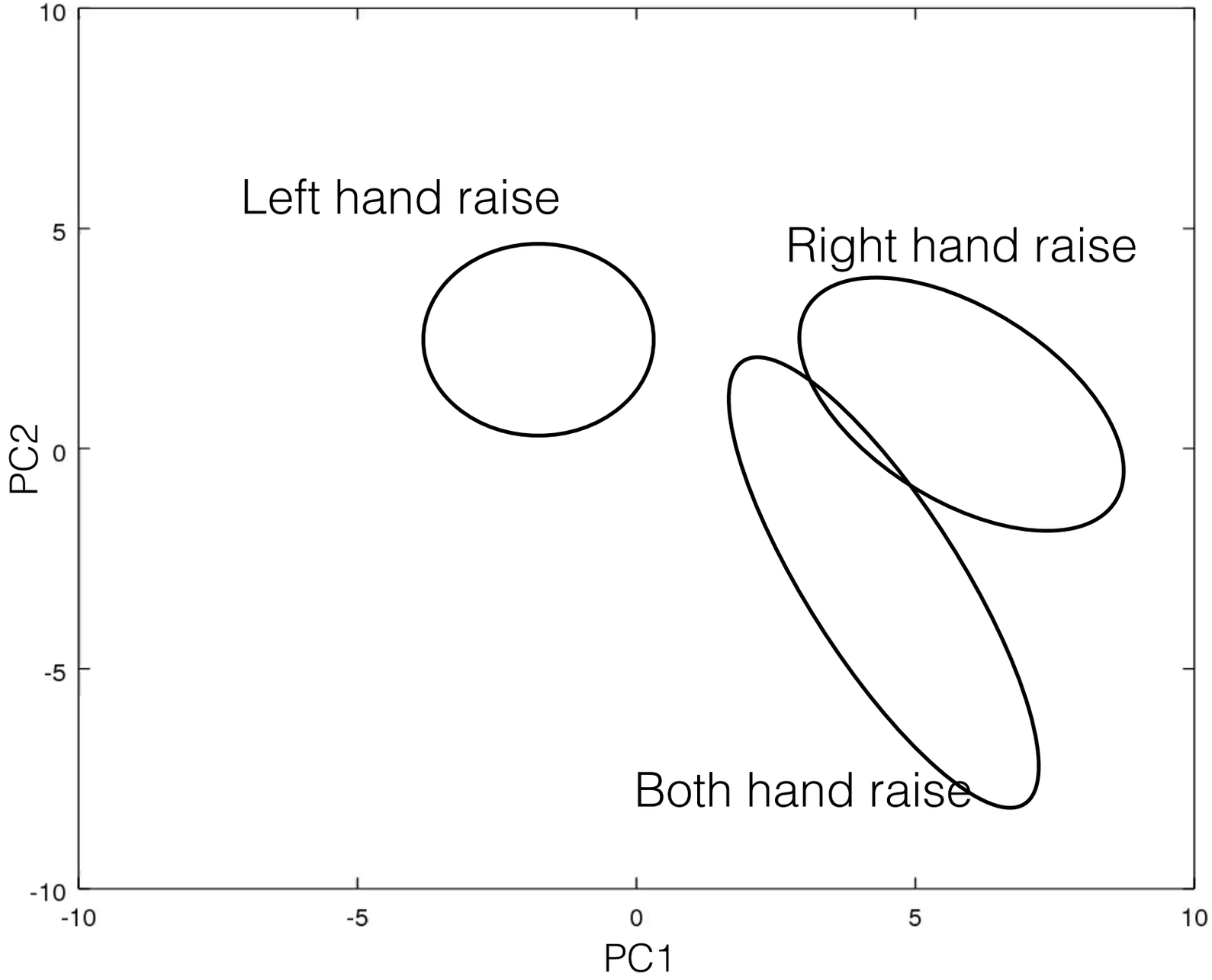


- Data set 3(right)

- Result: The model learned the mapping of visual data stream and corresponding joint info, instead of remembering the motion sequence

PCA of the Middle Layer Activation

- Red: Data set 1
- Green: Data set 2
- Blue: Data set 3



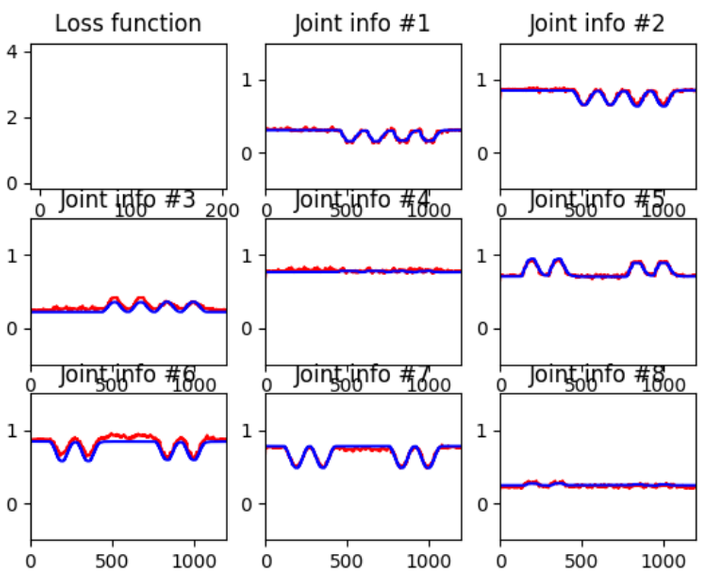
Back to Hypothesis

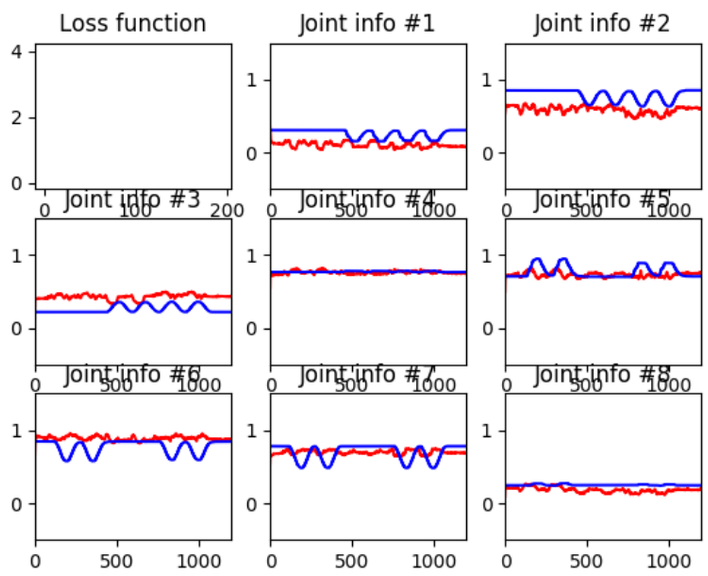
- Train the model with self sensorimotor mapping, test the same data set in 45° and 315° without any further training
  - Is the self sensorimotor experience enough to predict same motion in other perspectives?
- After the mapping, train the model to minimize the prediction error caused by visual input in 45° and 315°
  - Does the further training improve the joint prediction?
- Provide additional goal information when training different data sets
  - Assume same data set in different perspective share the same goal
  - Does this improve spatial perspective taking in higher angles?

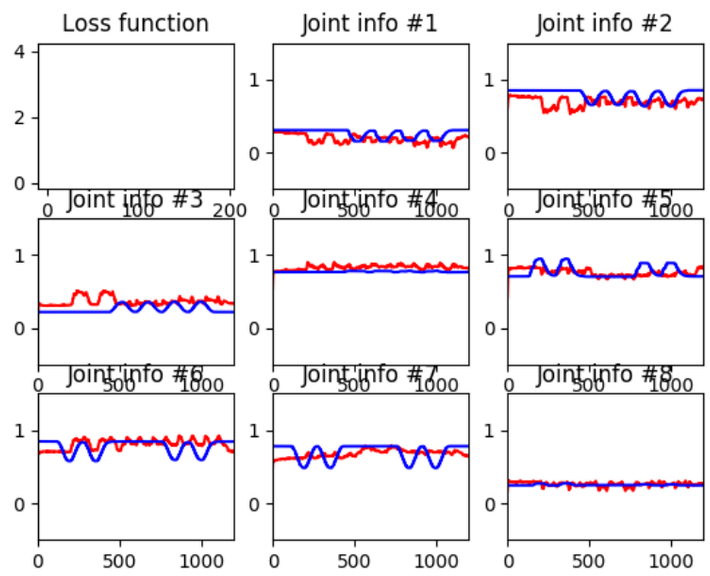
Note: The actual joint data is based on self motion sequence,
thus does not need to be aligned with the predicted sequence

Experiment 1

- Learn data set 1 of self, predict data set 1 of 45° and 315°







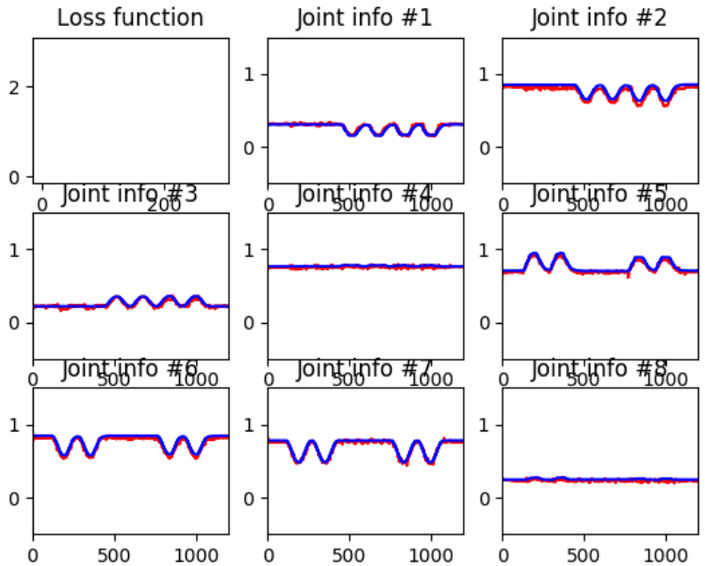
- Data set 1(self)
- Data set 1(45°)
- Data set 1(315°)

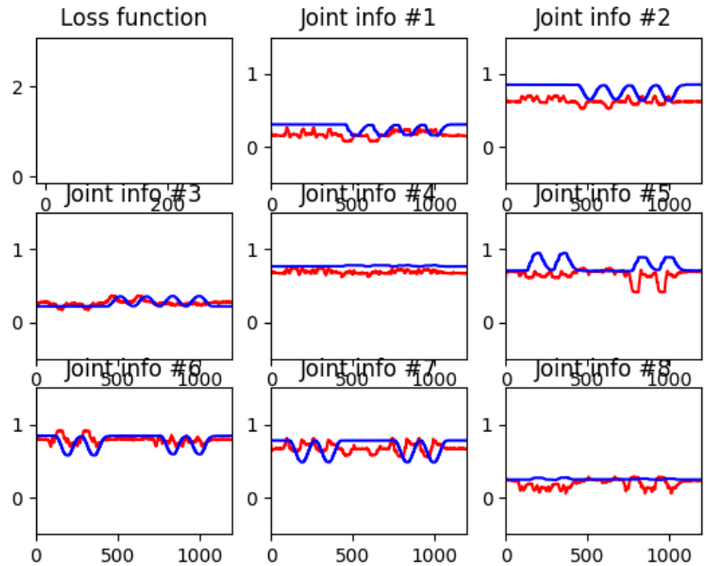
- Without any learning, the prediction does not show much recognition ability

Note: The actual joint data is based on self motion sequence,
thus does not need to be aligned with the predicted sequence

Experiment 2

- Learn data set 1 of self and 45°, predict data set 1 of self and 45°





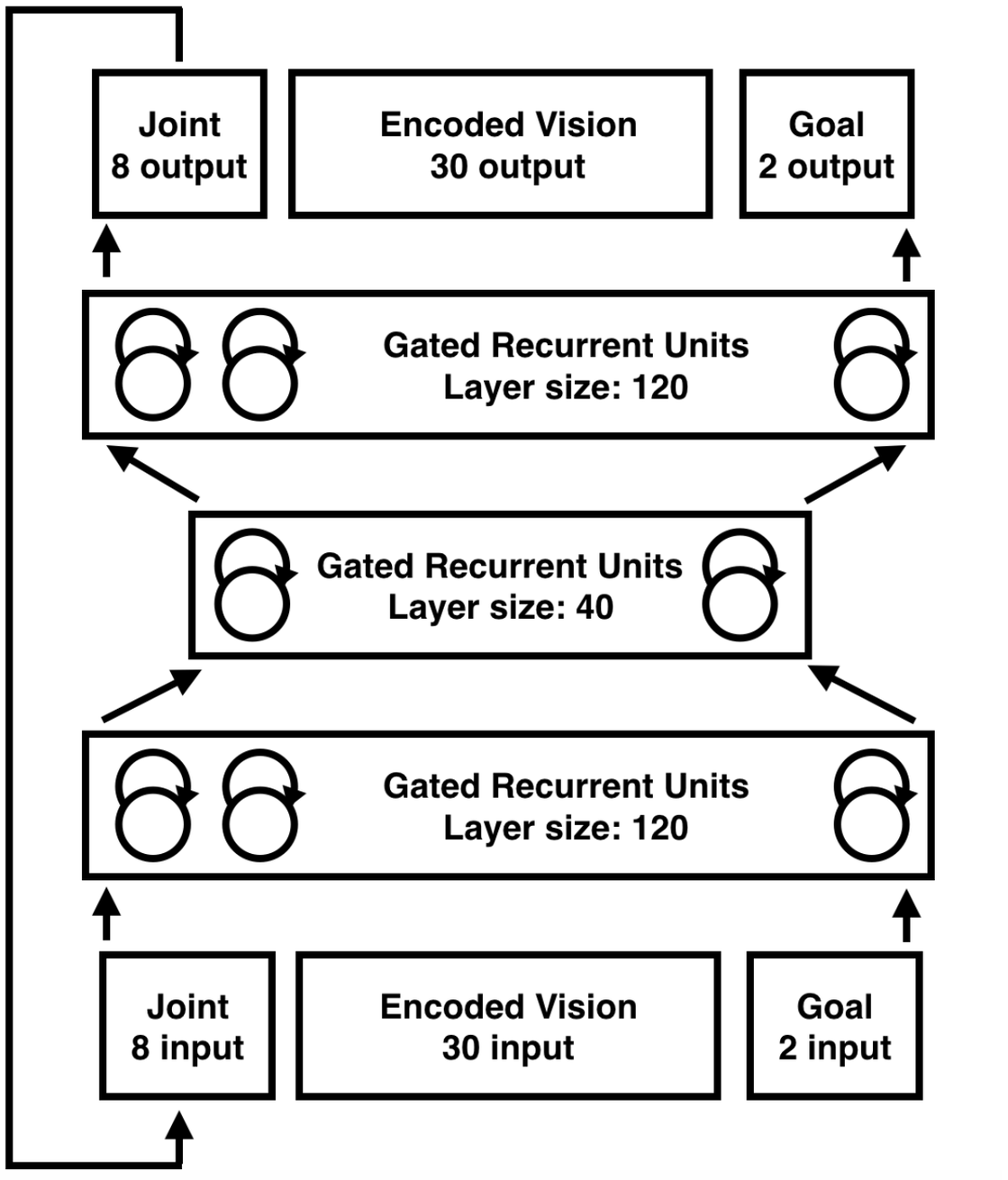
- Data set 1(self)
- Data set 1(45°)

- Even after learning, the prediction does not show much recognition ability
- Can’t see improvement

Note: GRU are not nodes, this graph is only for conceptual understanding
The GRU layer computed in the matrix manner

Model Modified

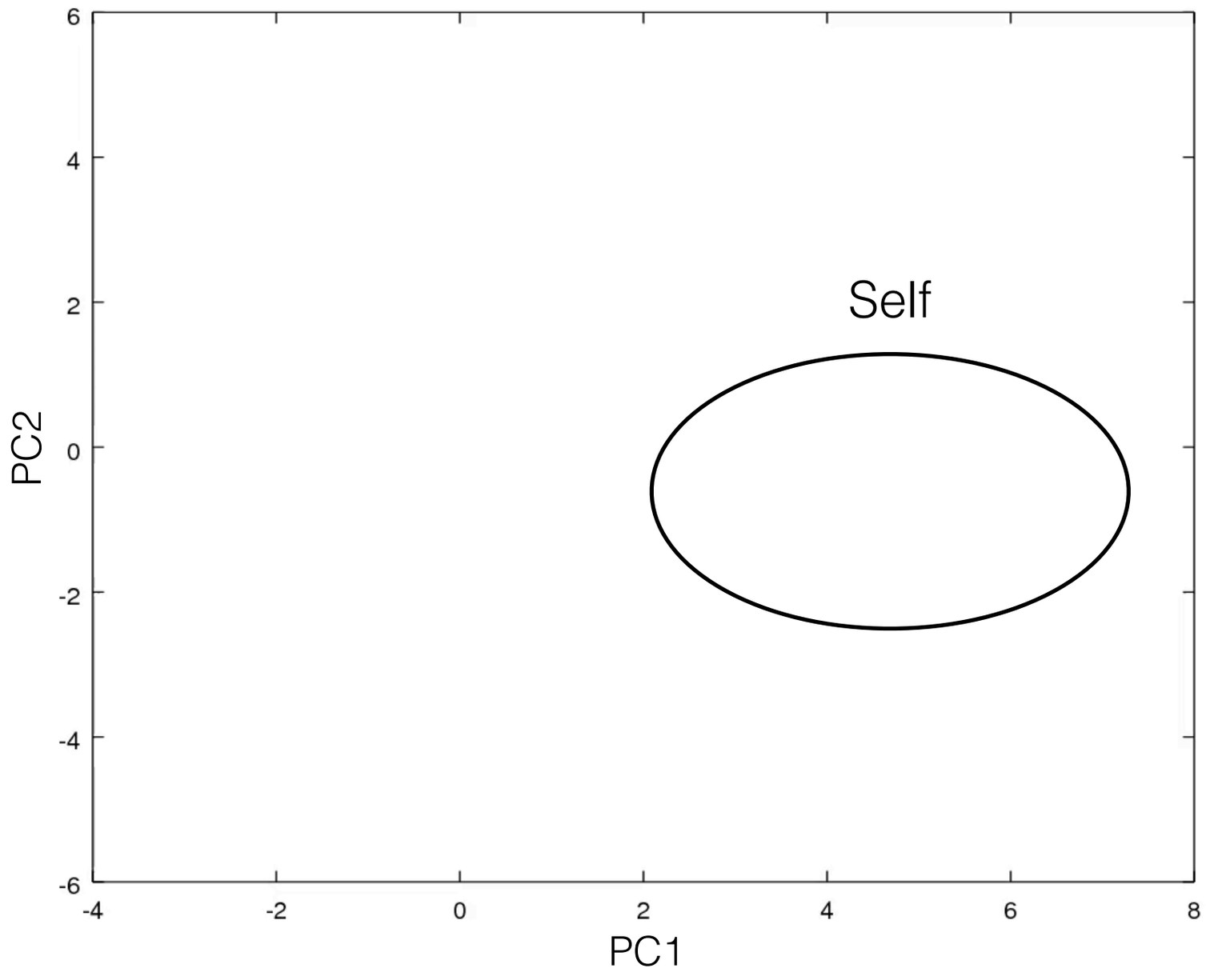
- Add two input dimensions indicating “goal”
  - assume each motion sequence(1, 2, 3) is associated with a different goal
  - assume each goal is encoded as two dimensional input
  - ‘00’: encode for motion 1
  - ‘10’: encode for motion 2
  - ‘01’: encode for motion 3
- Result: no improvement (screwed)



Note: Blue dotted line: self
red: 45°; green: 90°; blue line: 135°; yellow: 180°; magenta: 225°; cyan: 270°; black: 315°

Wait… How does the PCA look like?

- Train the model with data set 1 in all eight perspectives and test them



New Hypothesis

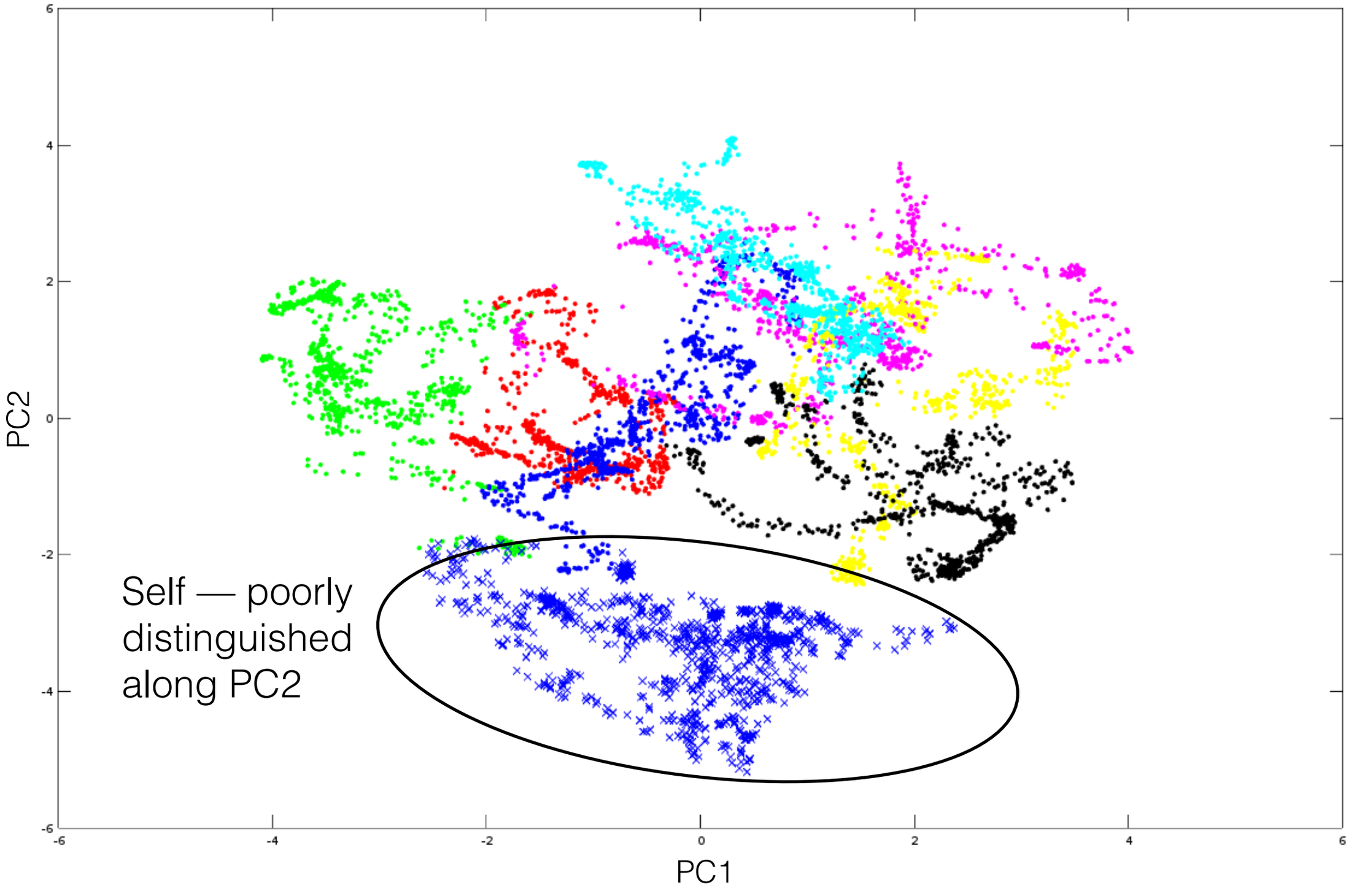
- Joint feedback played a crucial role in self-other recognition
- Human only have joint feedback when we are actually moving our body. Observing others may activate Mirror Neuron, but won't have joint feedback accordingly.
- The joint closed loop works as the Mirror Neuron System, which can distinguish whether its joint prediction was corrected by the real feedback
- In order to achieve that conclusion, need to eliminate the possibility that the visual data contains enough differences for the model to distinguish between self and others

New Experiments

- Repeat following experiments on each data sets:
  - First model: remove joint components, only learn to predict V(t+1) from V(t)
  - Second model: same model, don't provide joint feedback for self motions
  - Third model: same model, still have joint feed back for self
  - Train each model with data stream in all eight perspectives and test them

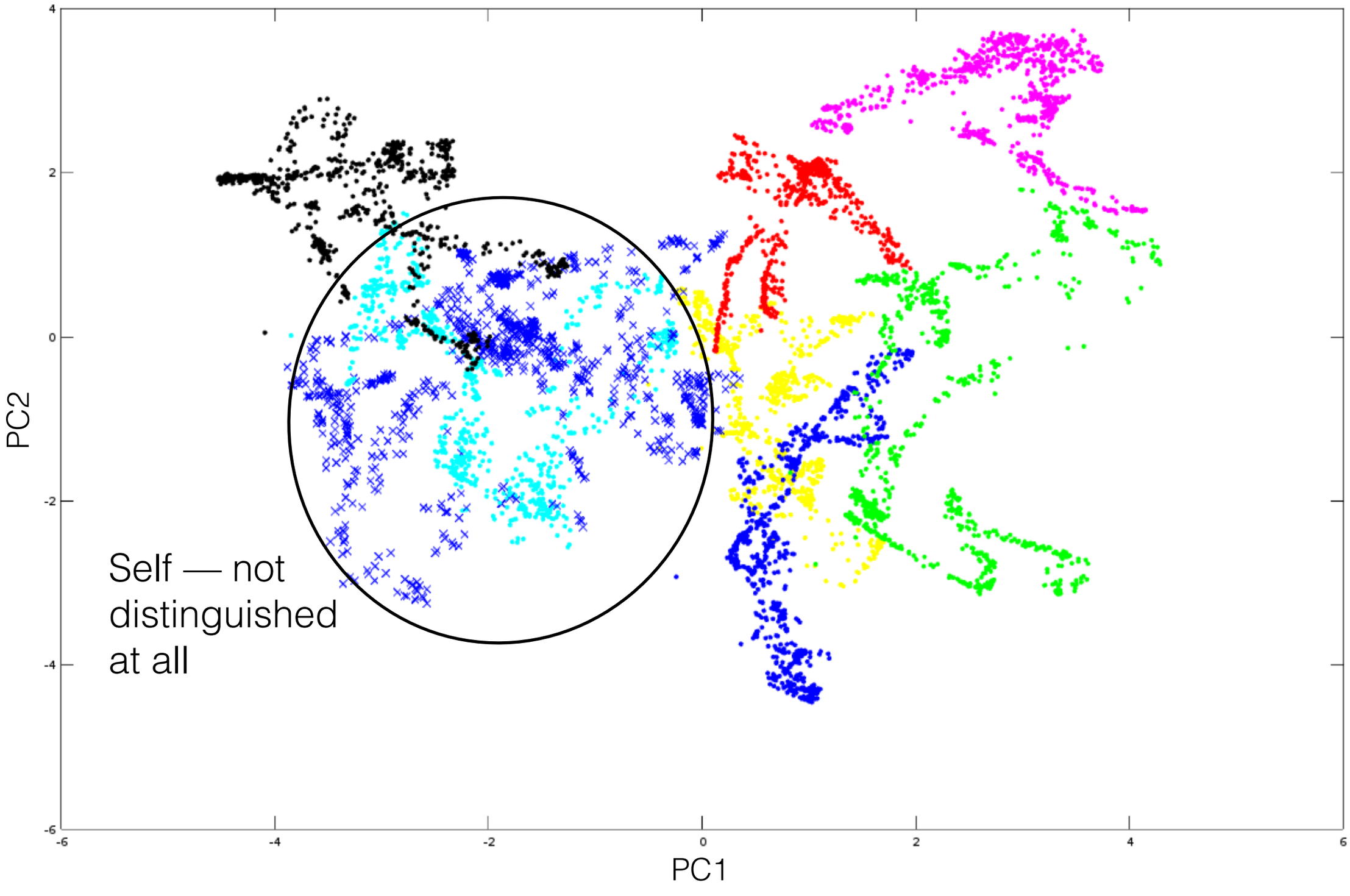
Note: Blue cross: self
red: 45°; green: 90°; blue dot: 135°; yellow: 180°; magenta: 225°; cyan: 270°; black: 315°

PCA on Data Set 1 — only vision prediction



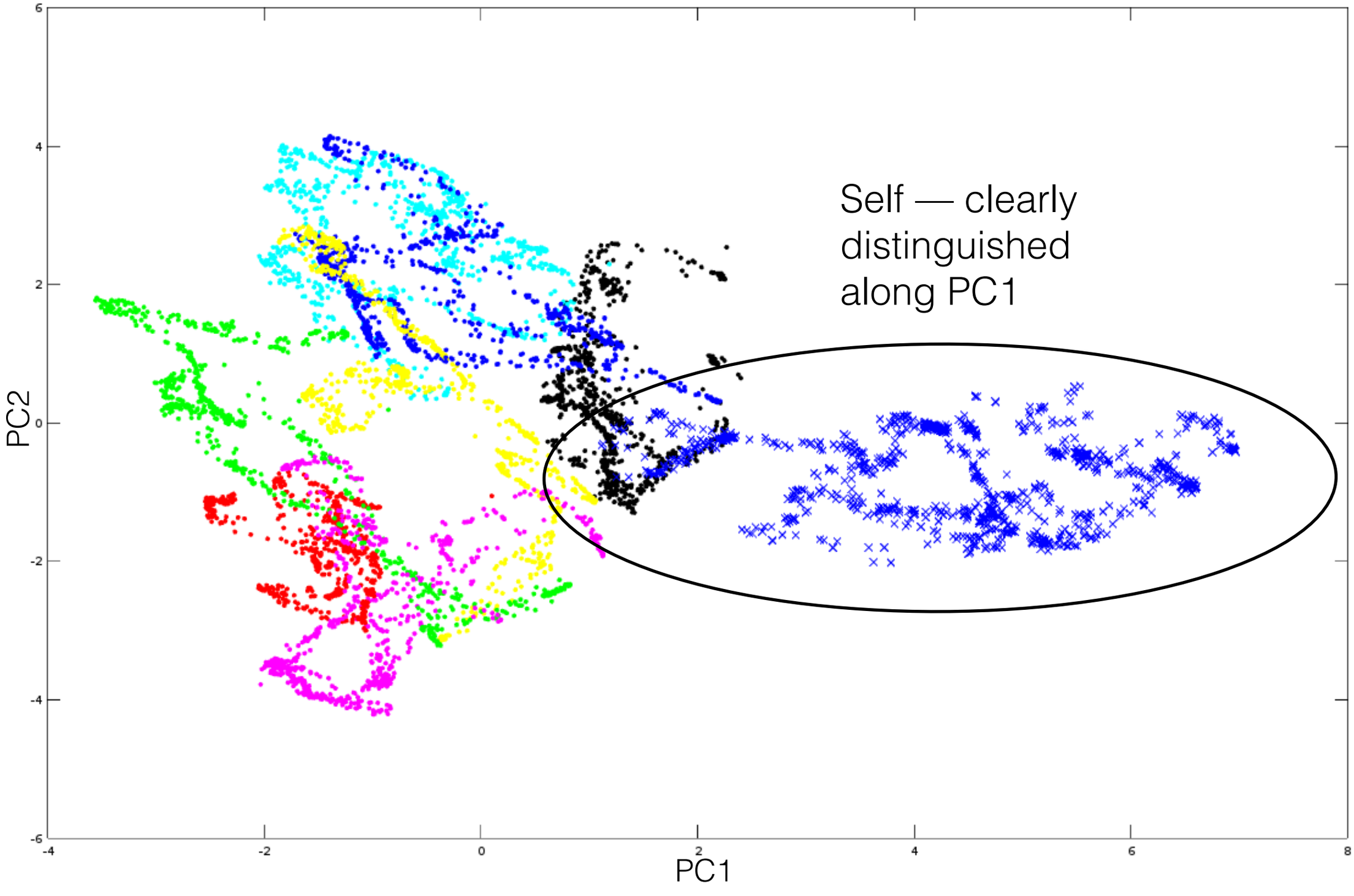
Note: Blue cross: self
red: 45°; green: 90°; blue dot: 135°; yellow: 180°; magenta: 225°; cyan: 270°; black: 315°

PCA on Data Set 1 — no joint feedback



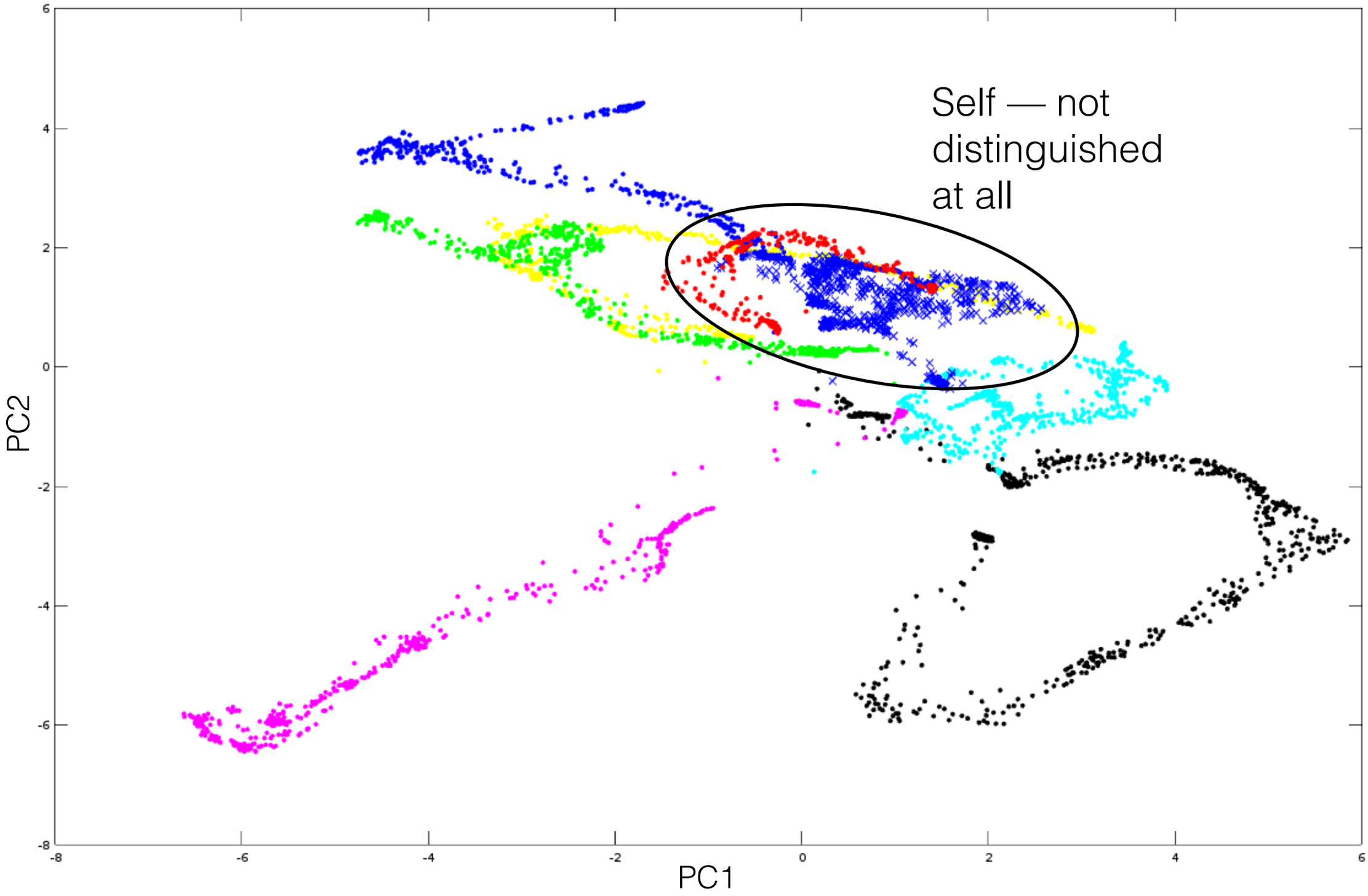
Note: Blue cross: self
red: 45°; green: 90°; blue dot: 135°; yellow: 180°; magenta: 225°; cyan: 270°; black: 315°

PCA on Data Set 1 — with joint feedback



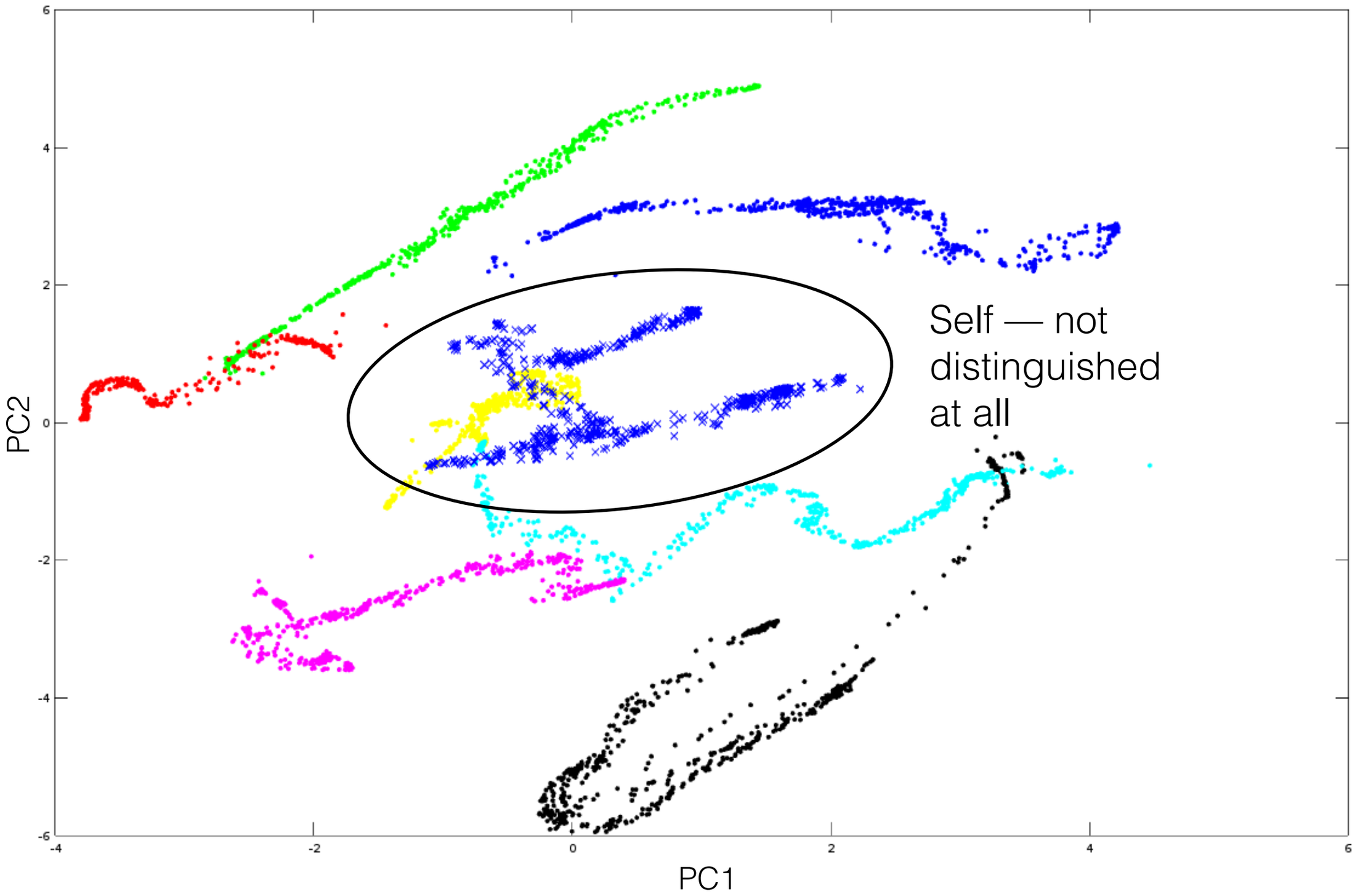
Note: Blue cross: self
red: 45°; green: 90°; blue dot: 135°; yellow: 180°; magenta: 225°; cyan: 270°; black: 315°

PCA on Data Set 2 — only vision prediction



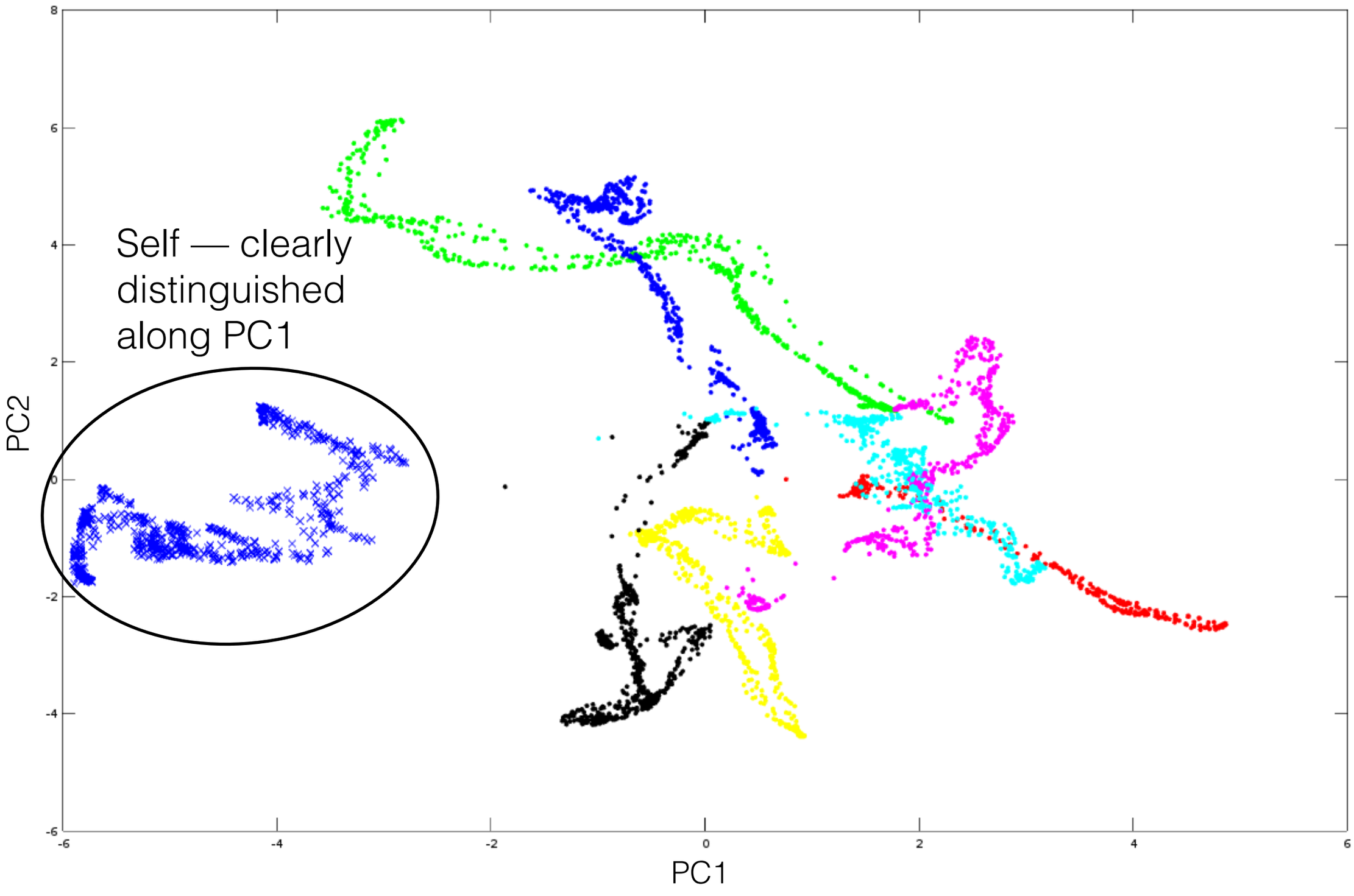
Note: Blue cross: self
red: 45°; green: 90°; blue dot: 135°; yellow: 180°; magenta: 225°; cyan: 270°; black: 315°

PCA on Data Set 2 — no joint feedback



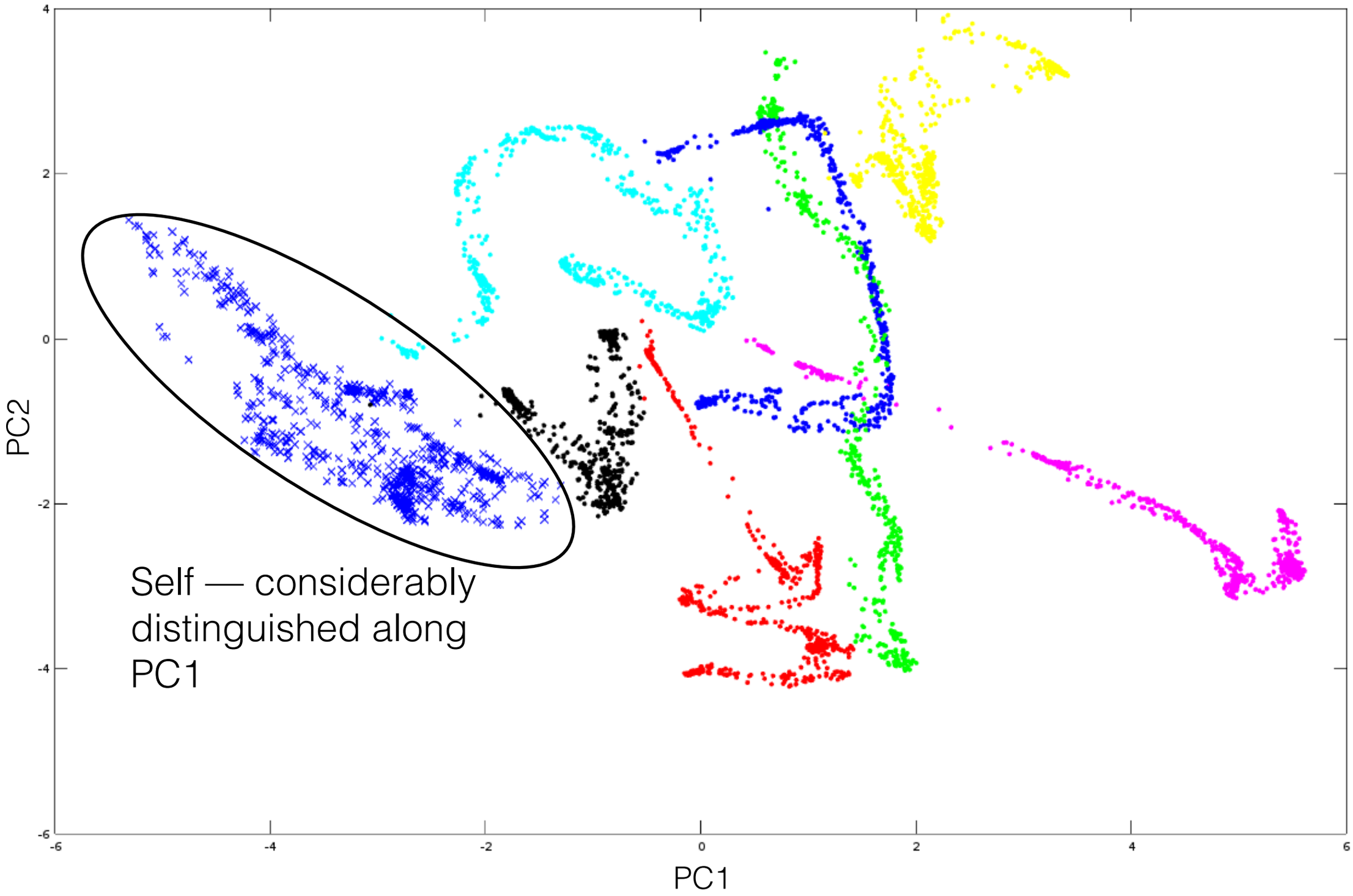
Note: Blue cross: self
red: 45°; green: 90°; blue dot: 135°; yellow: 180°; magenta: 225°; cyan: 270°; black: 315°

PCA on Data Set 2 — with joint feedback



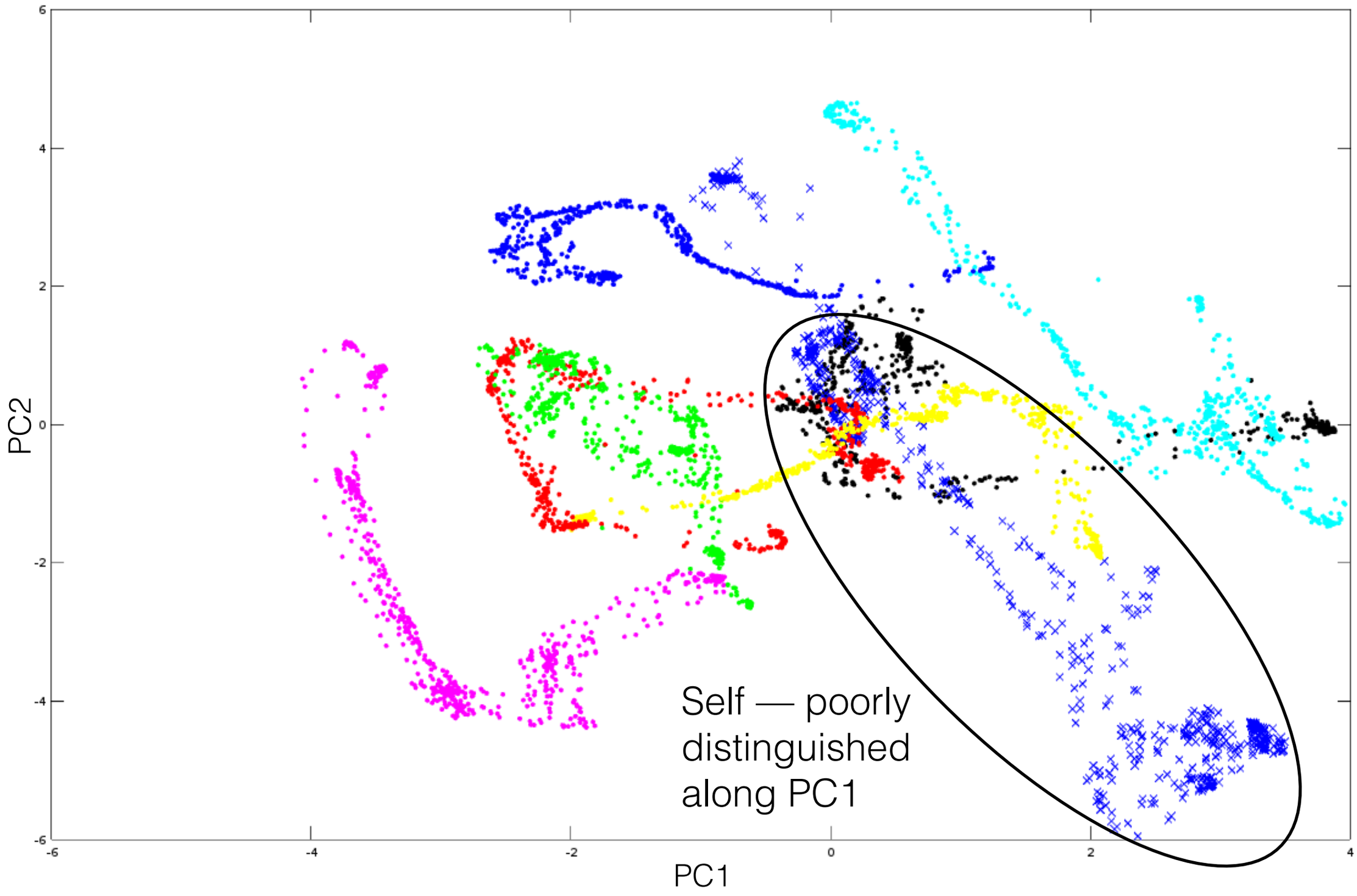
Note: Blue cross: self
red: 45°; green: 90°; blue dot: 135°; yellow: 180°; magenta: 225°; cyan: 270°; black: 315°

PCA on Data Set 3 — only vision prediction



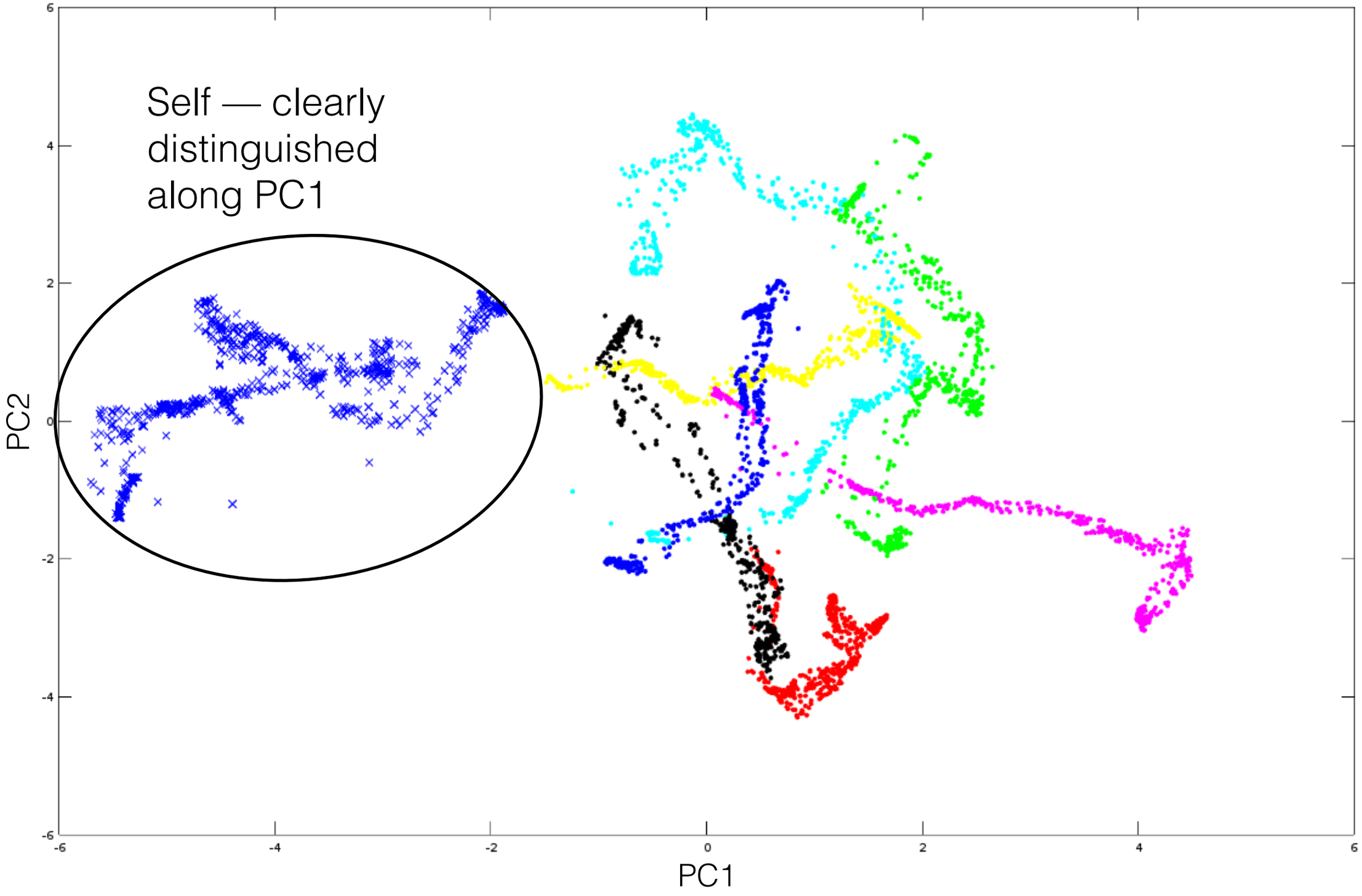
Note: Blue cross: self
red: 45°; green: 90°; blue dot: 135°; yellow: 180°; magenta: 225°; cyan: 270°; black: 315°

PCA on Data Set 3 — no joint feedback



Note: Blue cross: self
red: 45°; green: 90°; blue dot: 135°; yellow: 180°; magenta: 225°; cyan: 270°; black: 315°

PCA on Data Set 3 — with joint feedback



Conclusion From the Result (and more)

- Joint feedback helped the model to distinguish if the input visual stream is the motion of self or the motion of others
- But why? How does the model acquire this distinction and represent it in the middle layer activation?
- Does this result still hold true if there are multiple agents, including self, acting in the same environment (data set)?
- How can we use this distinction in the representation to help robot to develop the concept of self and non-self?
- Source code and data on my github: https://github.com/shzhangyihan/SPTNao
  - currently messy as hell
  - the model implemented using tensorflow might be useful

Finally

- Thank you guys so much for the time being here with me, your supports are really appreciated!!
- Thank you Nagai Sensei for all the inspirations and guidance!!
- This place is AWESOME!!