Doing Graduate Research

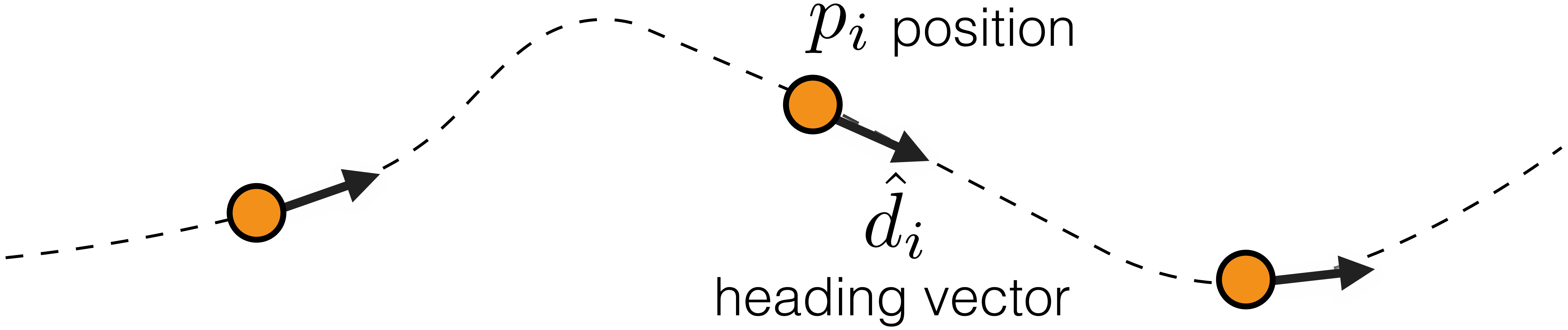
9.9.2022

What’s most challenging about research?

Find a great idea…

Example: My 2nd Ph.D. Project

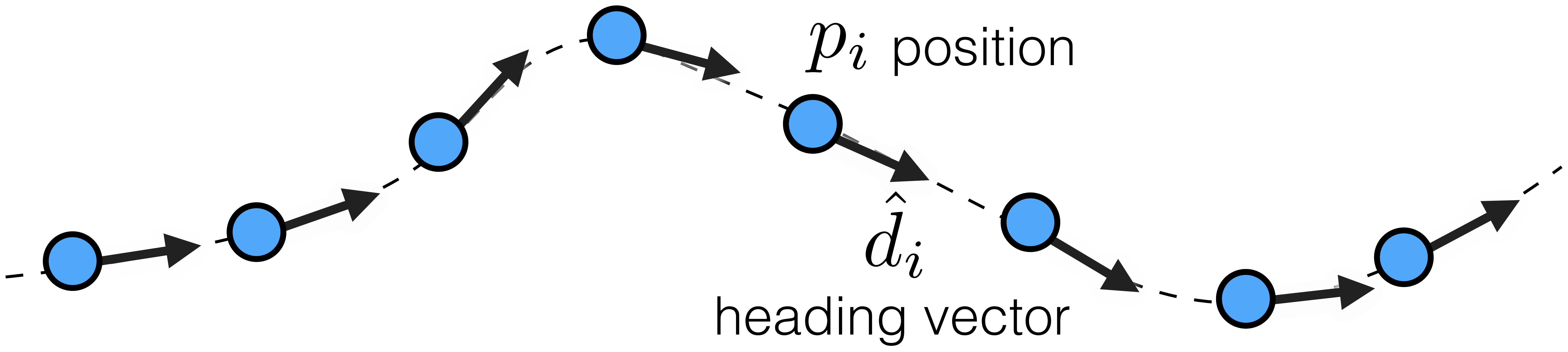
$$\tau = \langle (p_1, \hat{d}_1), \ldots, (p_{|\tau|}, \hat{d}_{|\tau|}) \rangle$$



Unorganized sparse trajectories -> dense trajectories

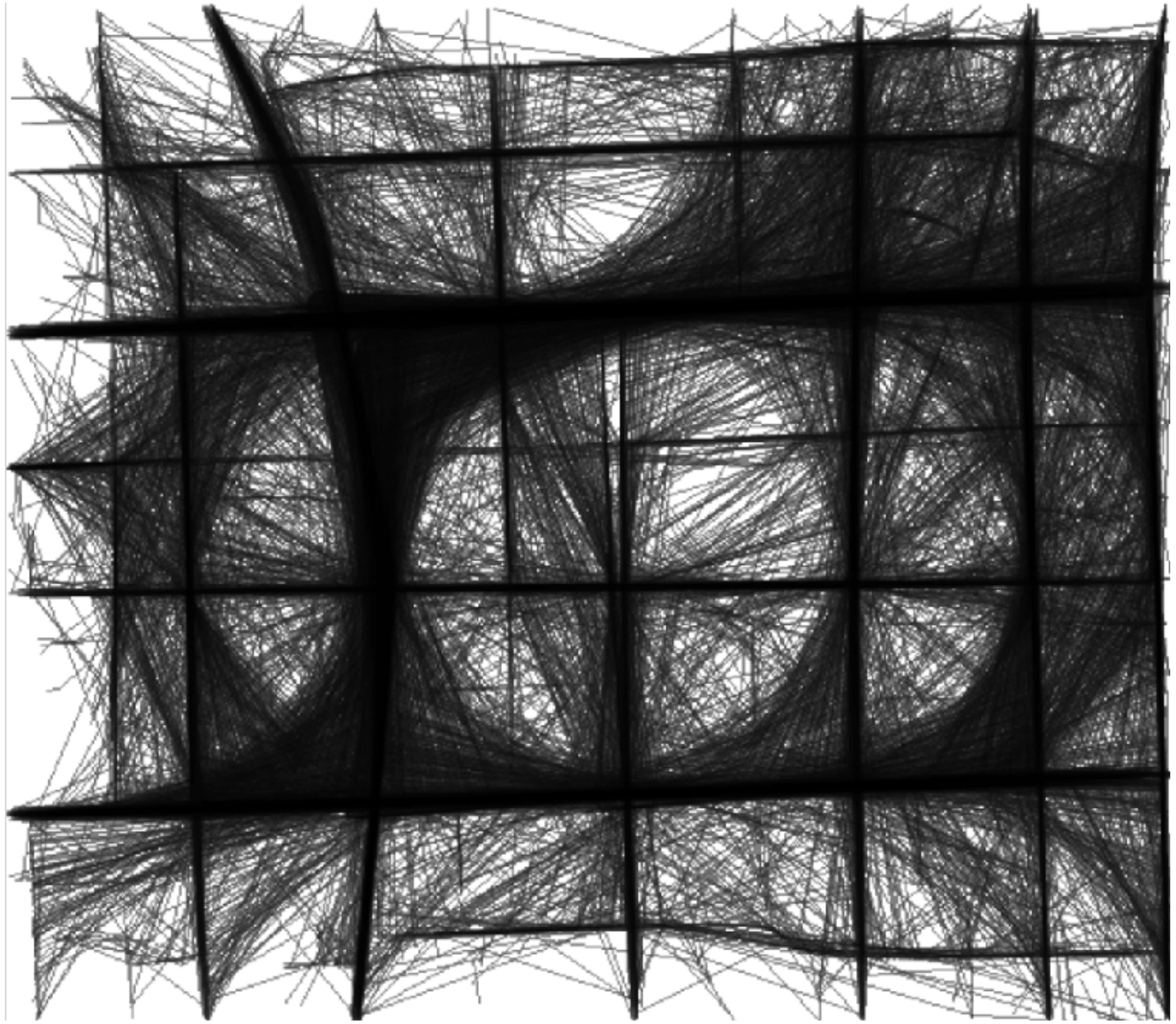
Example: My 2nd Ph.D. Project

$$\tau = \langle (p_1, \hat{d}_1), \ldots, (p_{|\tau|}, \hat{d}_{|\tau|}) \rangle$$



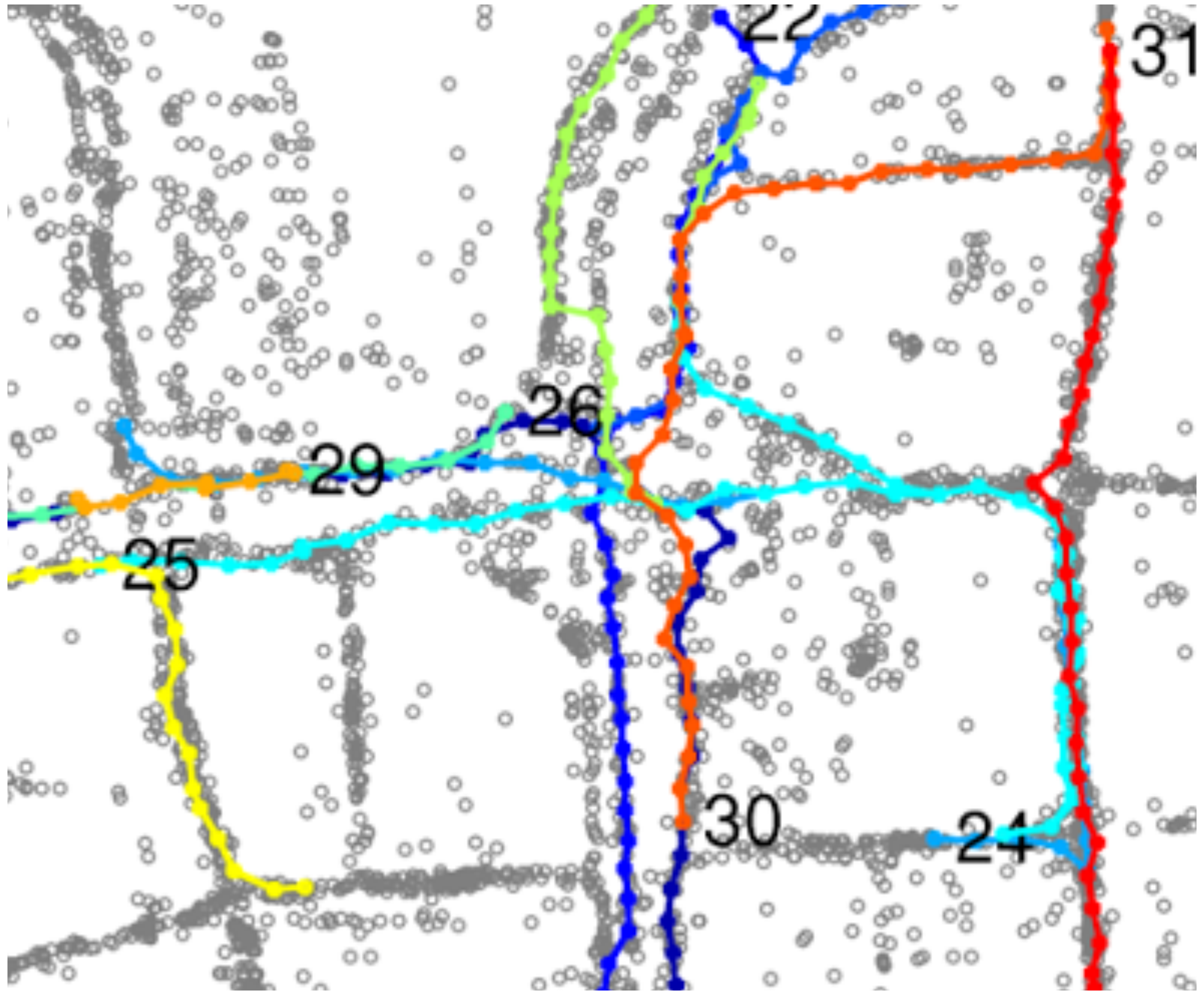
Unorganized sparse trajectories -> dense trajectories

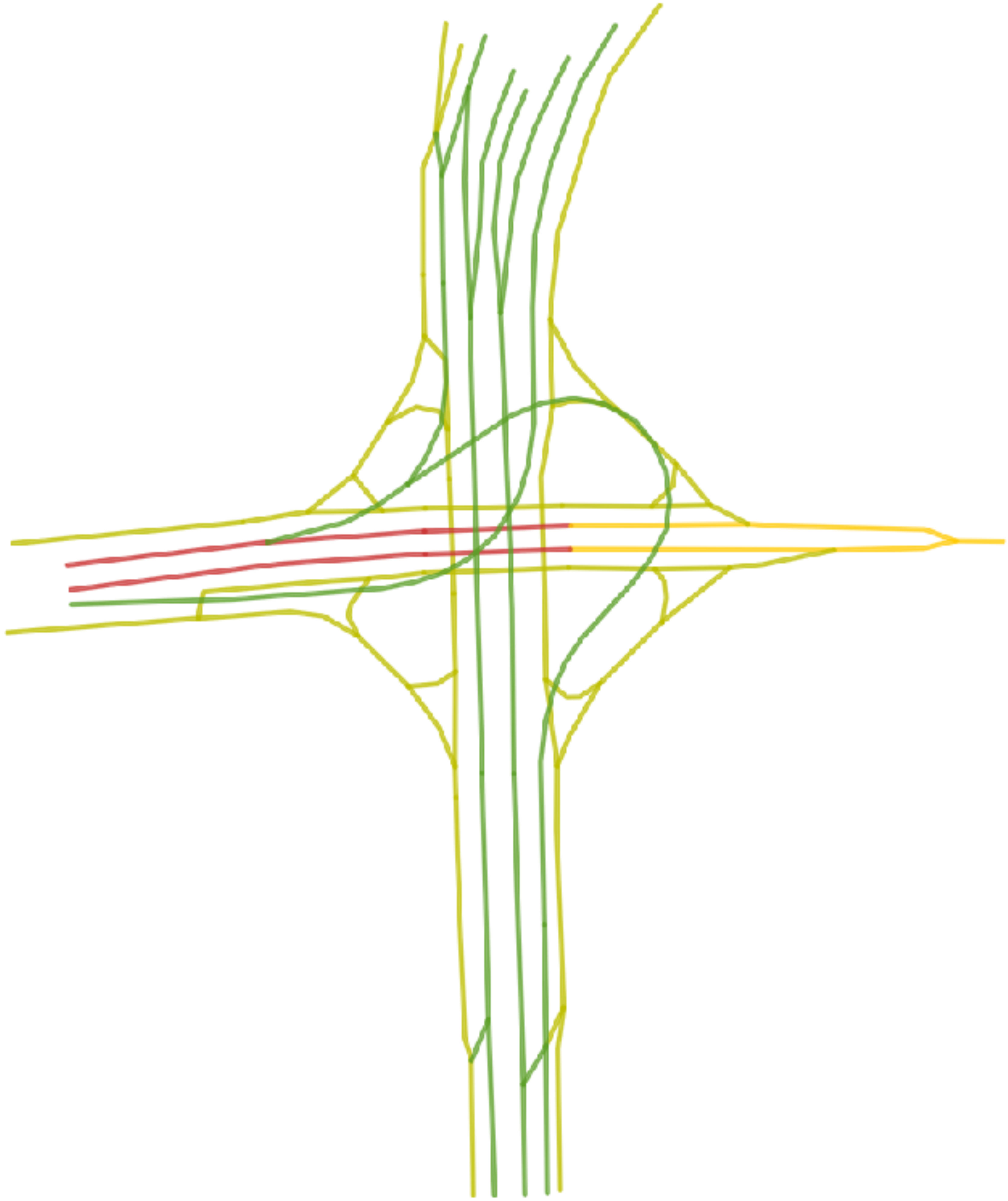
Example: My 2nd Ph.D. Project



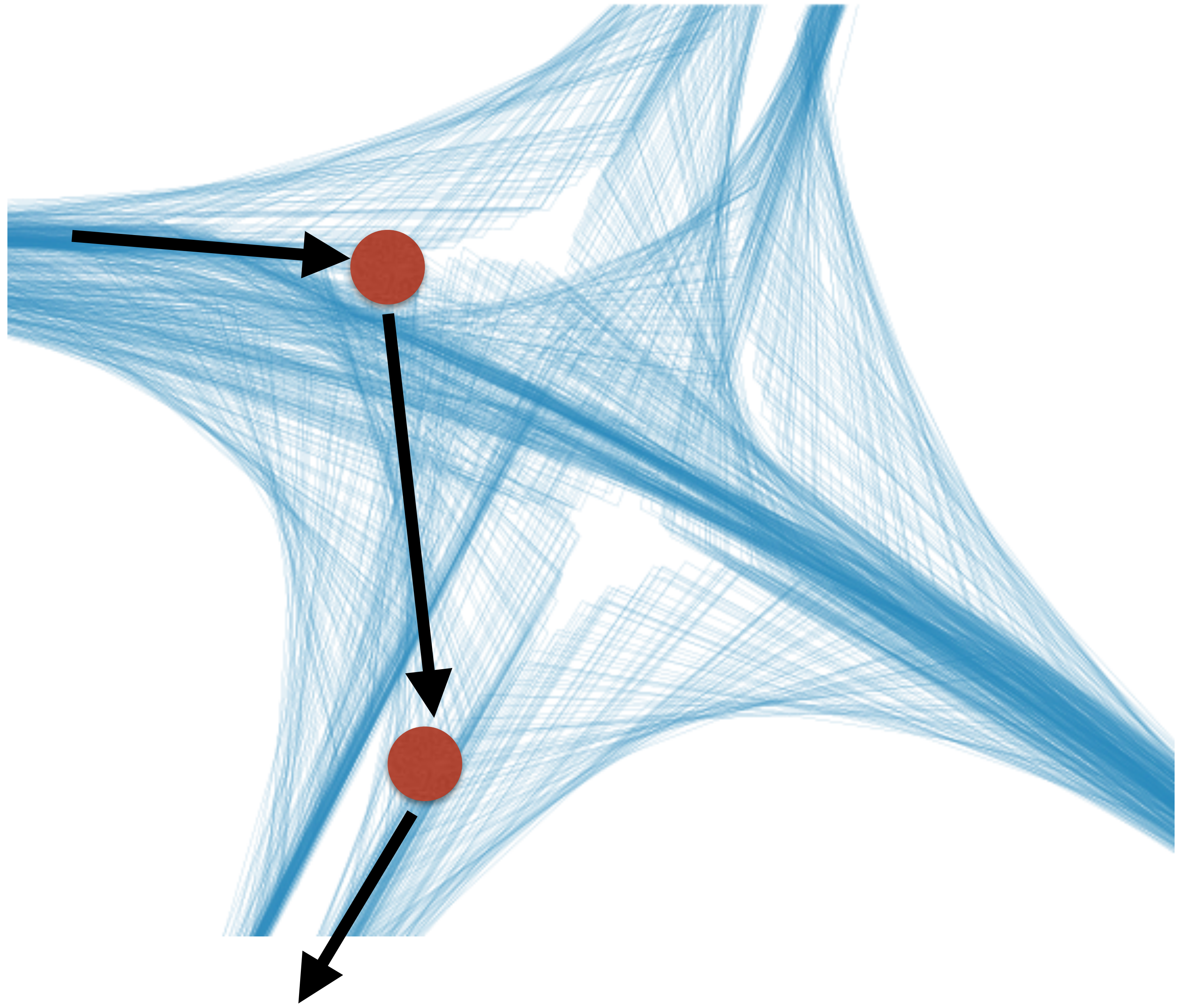
Unorganized sparse trajectories -> dense trajectories

Previous work doesn’t work at junctions

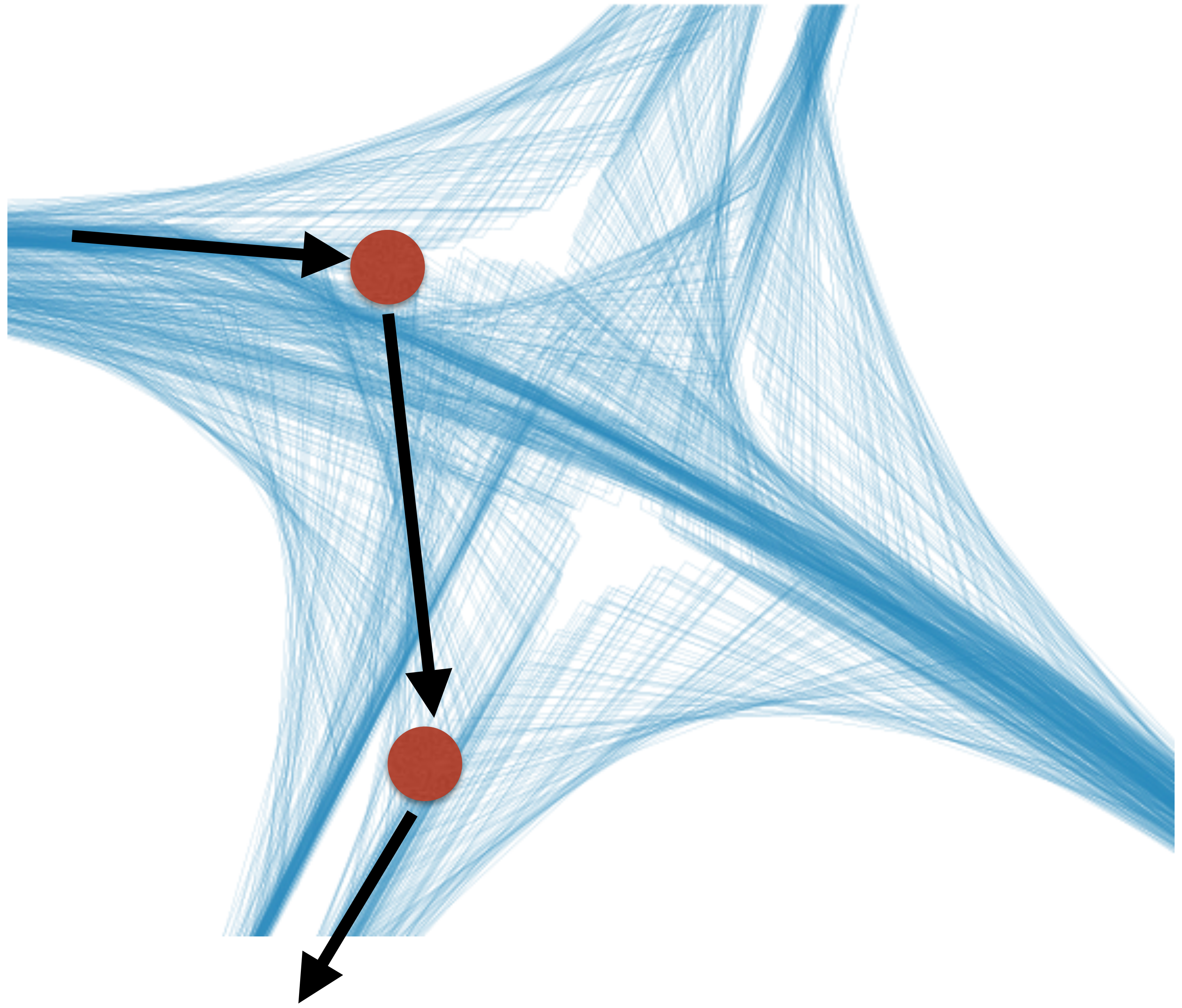


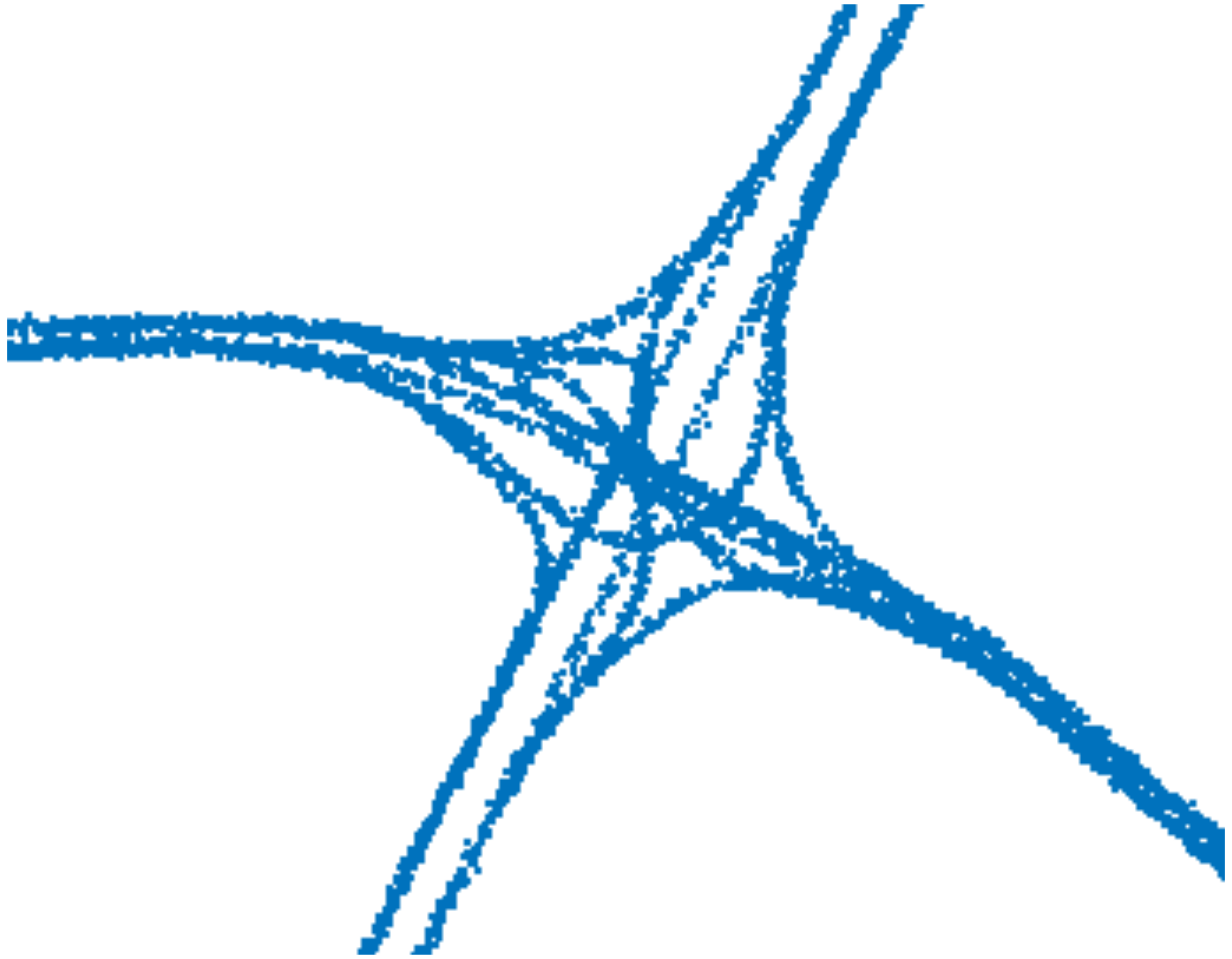


Is it possible?



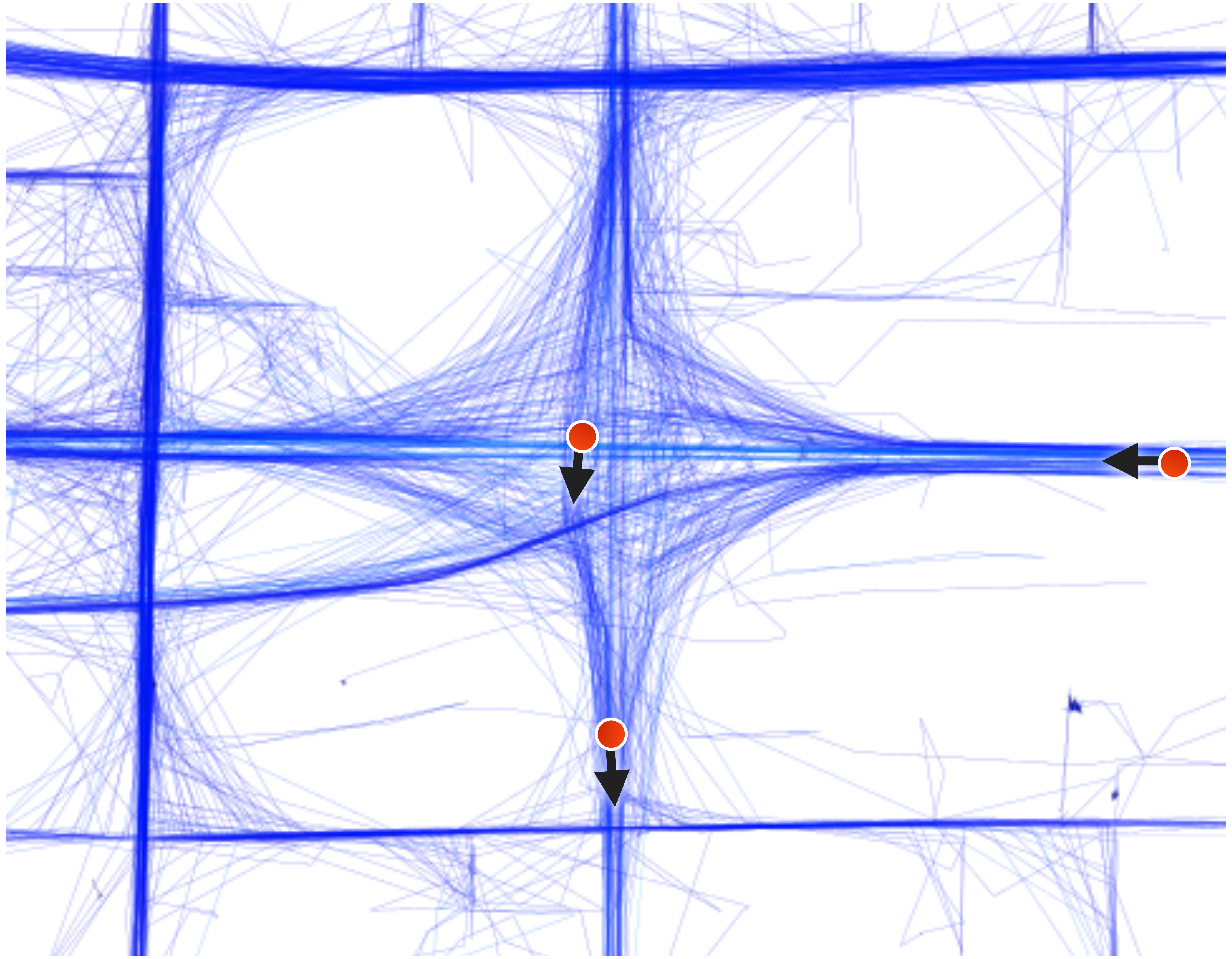
Is it possible?





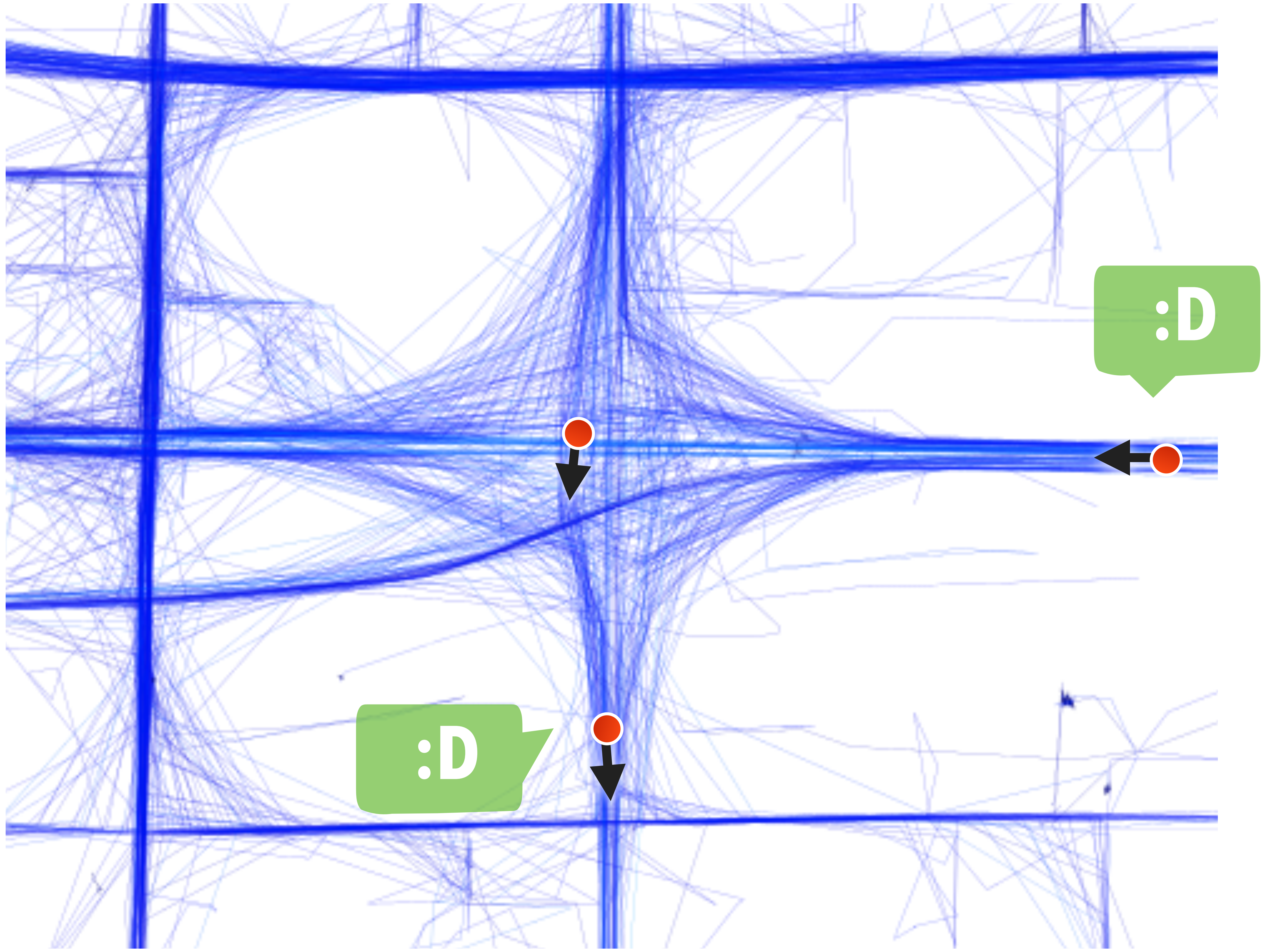
Looks easy for human eyes
(moving)

Our Intuition



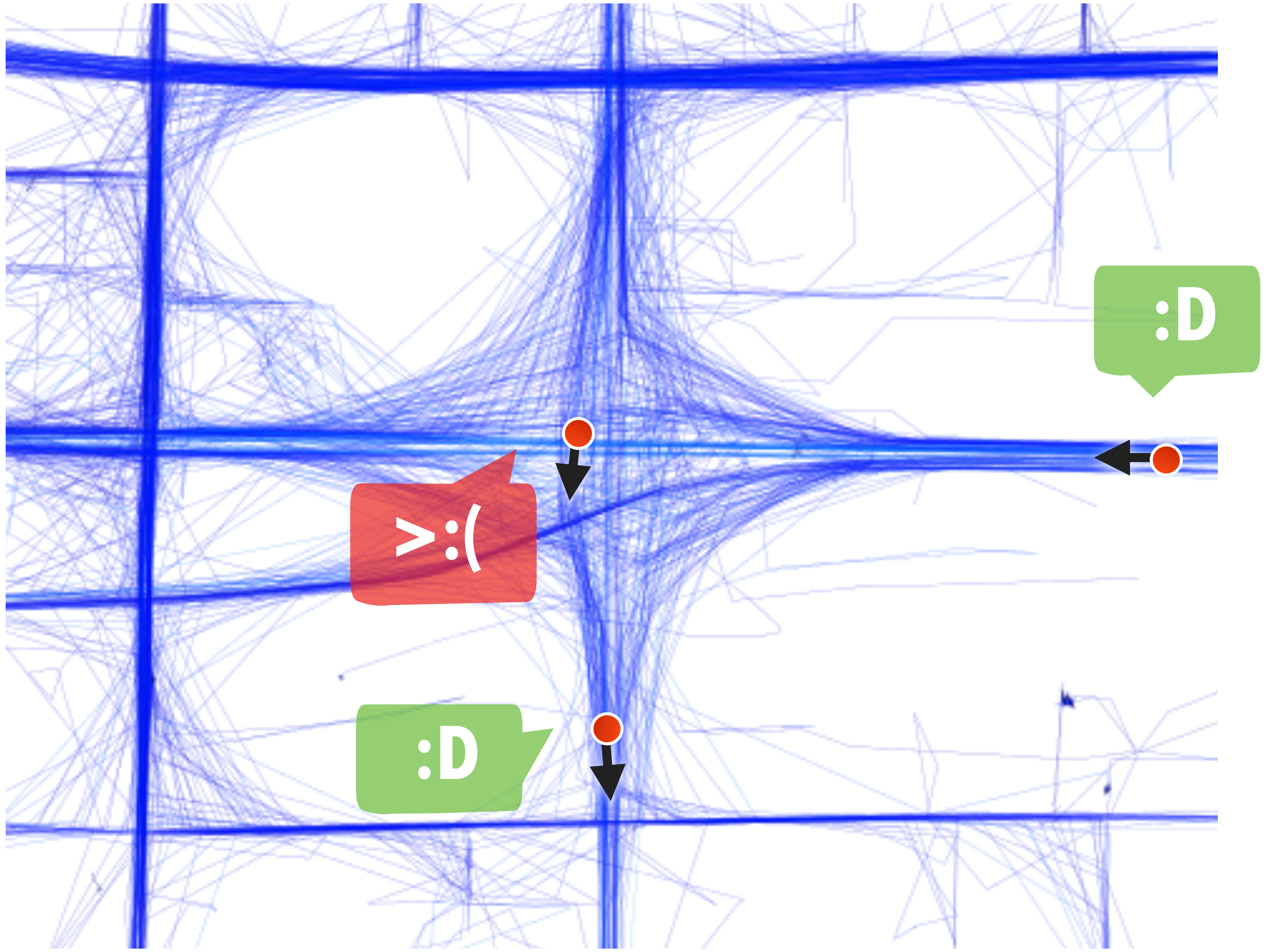
Key idea: Infer cluster knowledge in the **junction** region from **non-junction** regions

Our Intuition



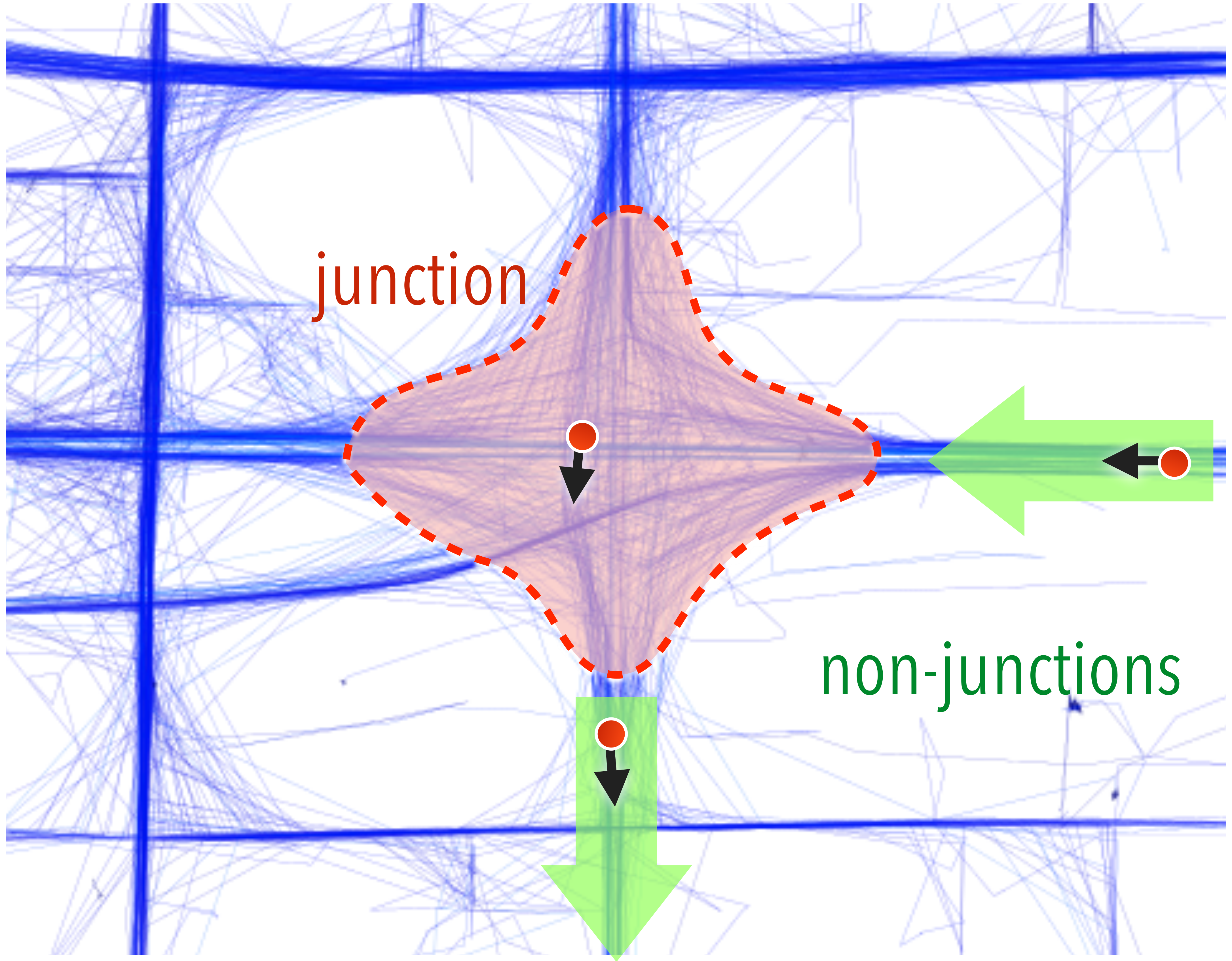
Key idea: Infer cluster knowledge in the **junction** region from **non-junction** regions

Our Intuition



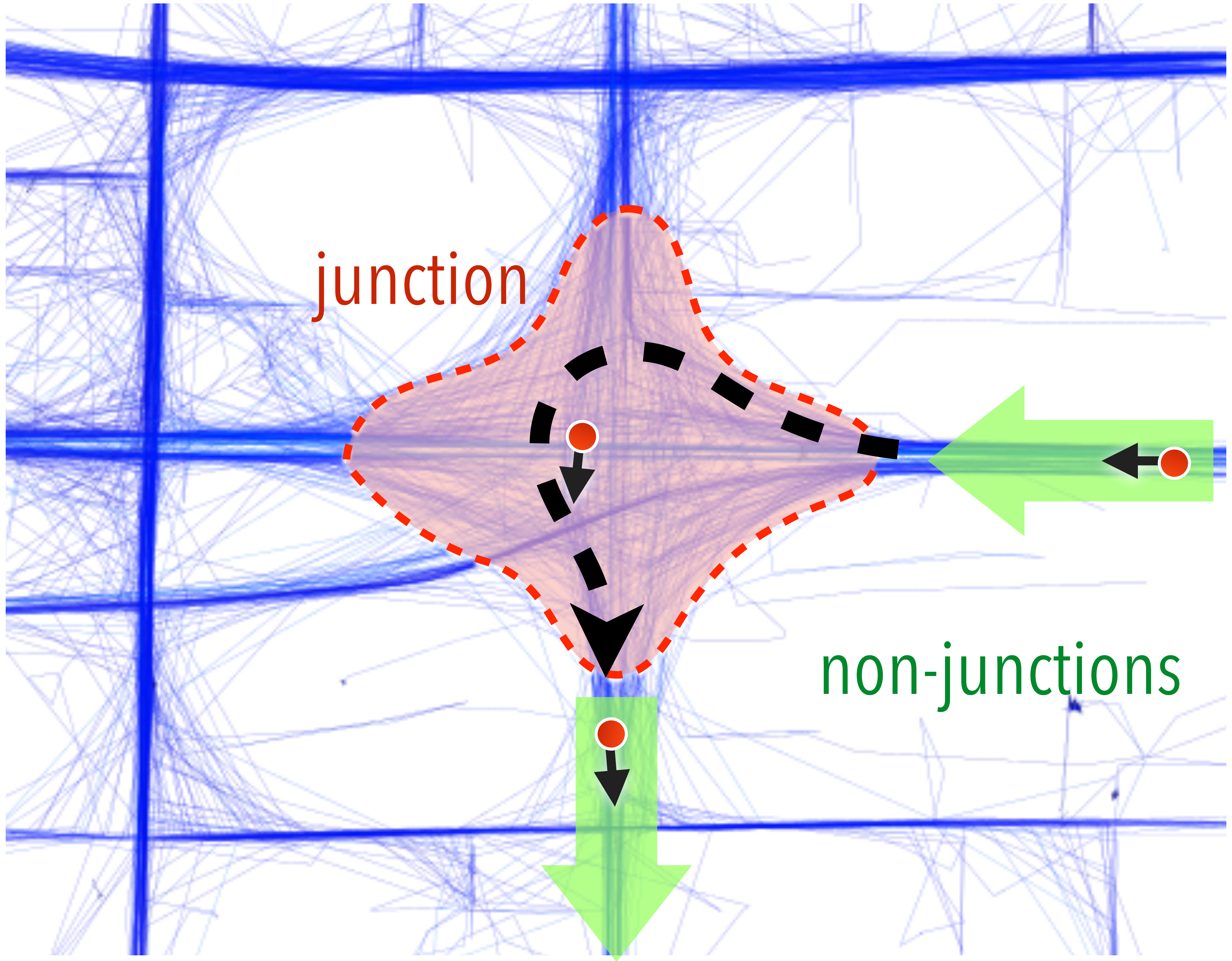
Key idea: Infer cluster knowledge in the **junction** region from **non-junction** regions

Our Intuition



Key idea: Infer cluster knowledge in the **junction** region from **non-junction** regions

Our Intuition

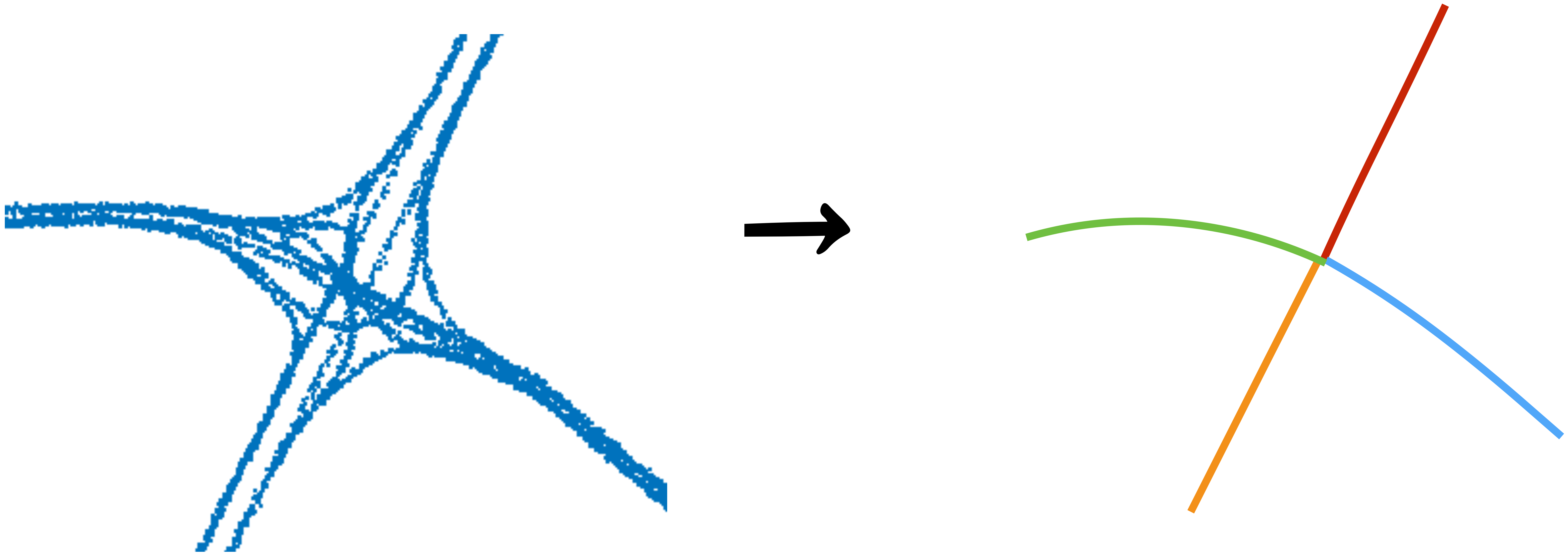


Key idea: Infer cluster knowledge in the **junction** region from **non-junction** regions

1. Aim high, trust your intuition

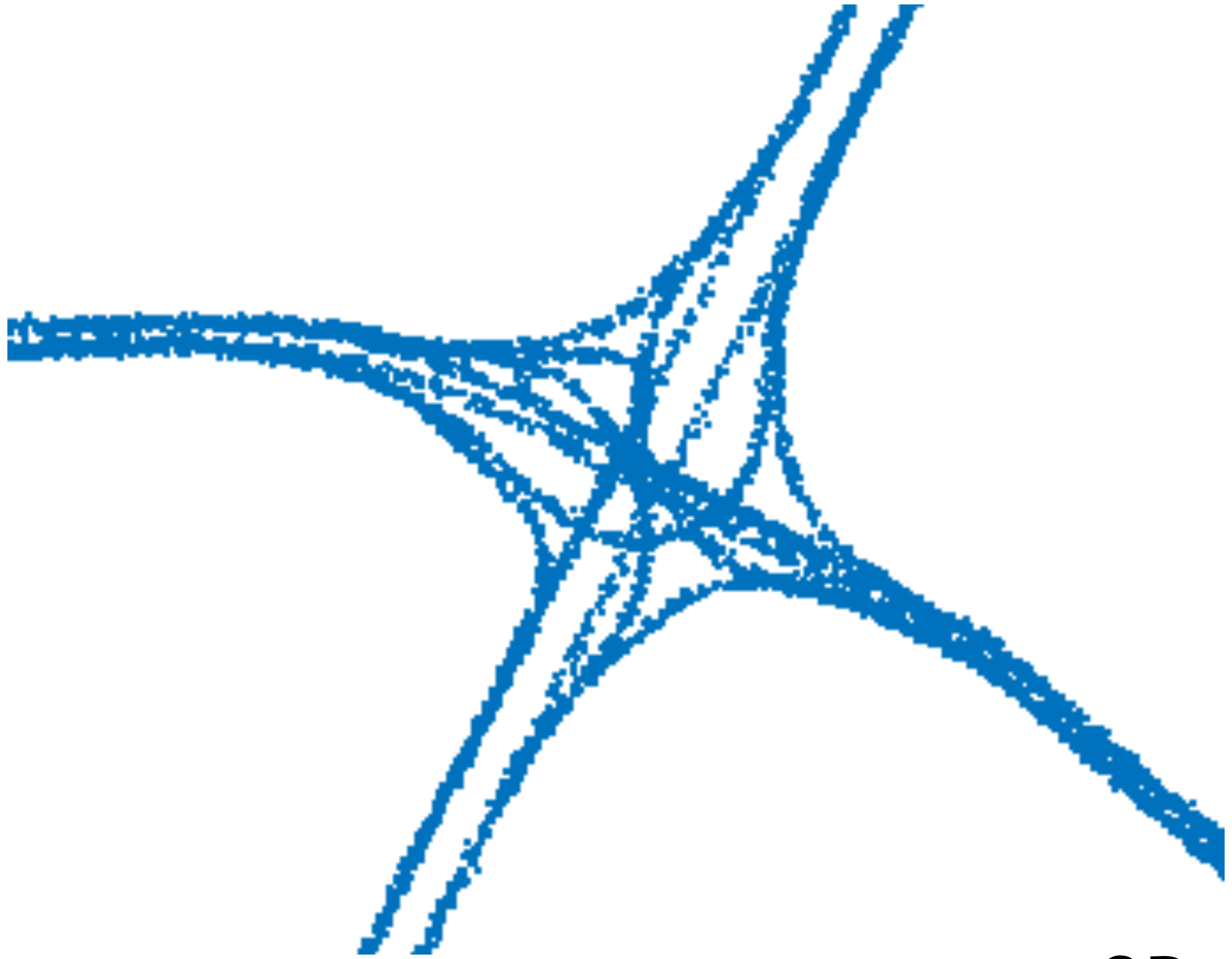
From intuition to a solution

How to extract junction branches?



From intuition to a solution

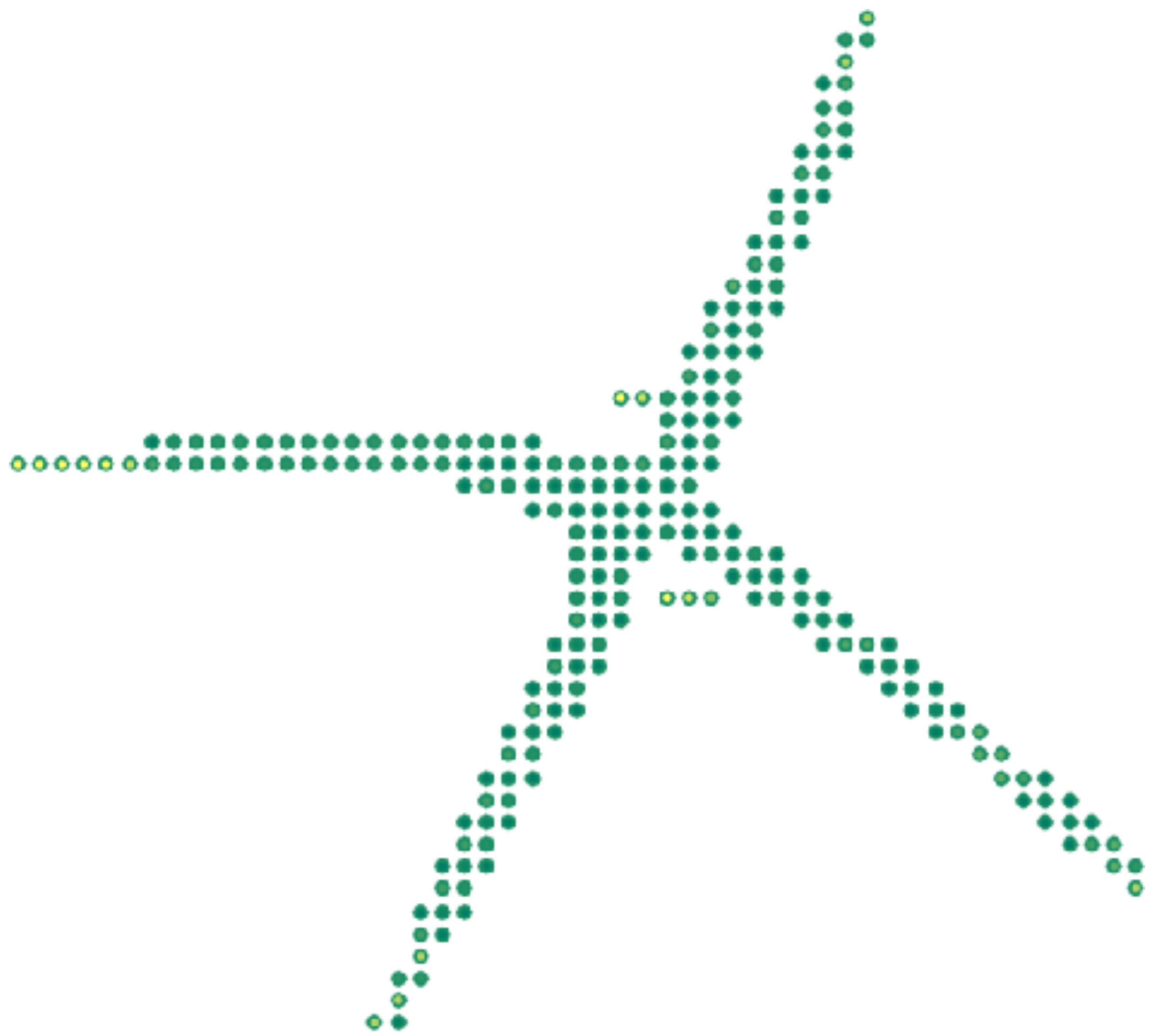
How to extract junction branches?



(a) A deer model. (b) Raw scan. (c) L_1-medial skeleton.

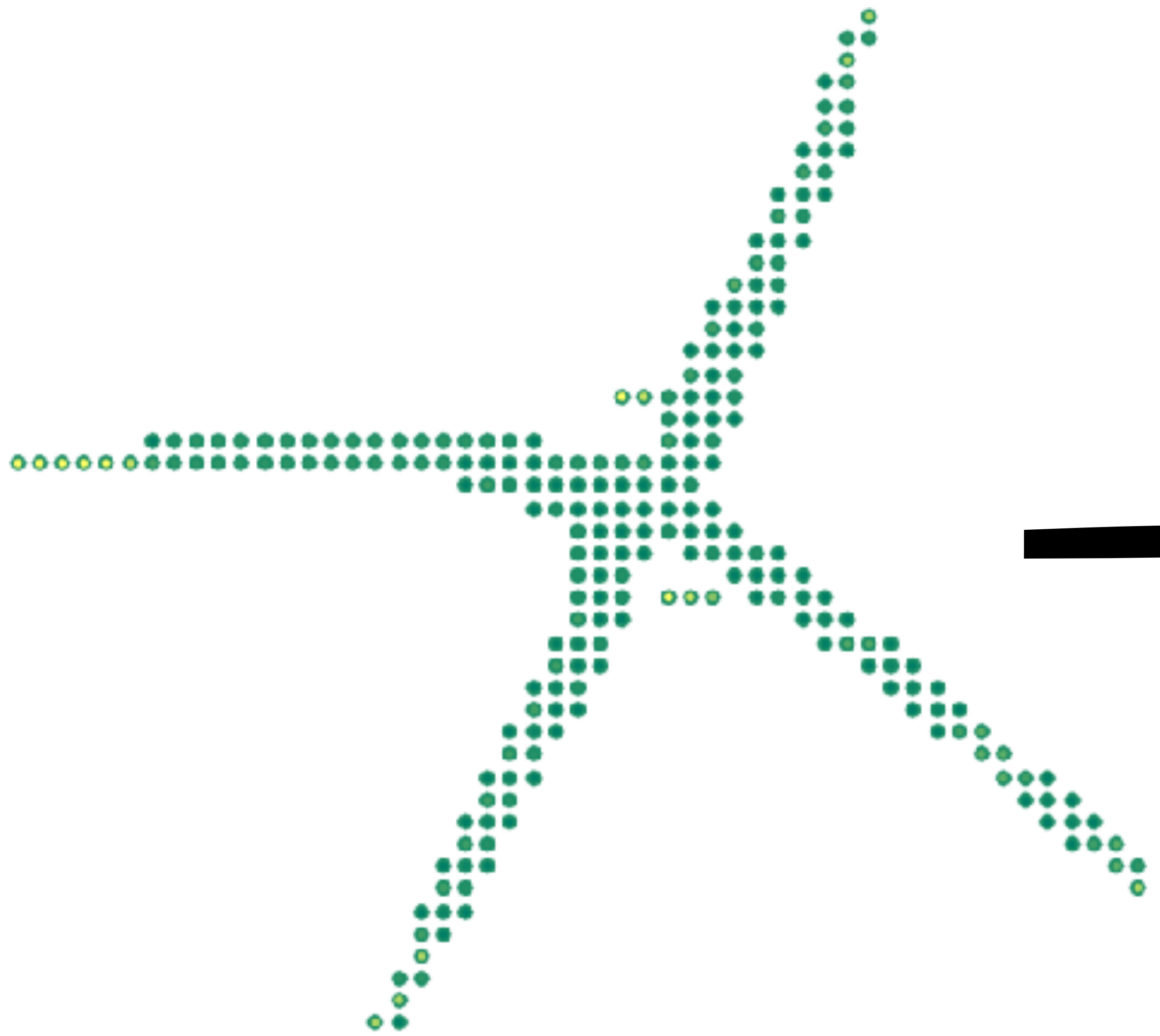
3D skeleton extraction from point cloud (2014)

From intuition to a solution

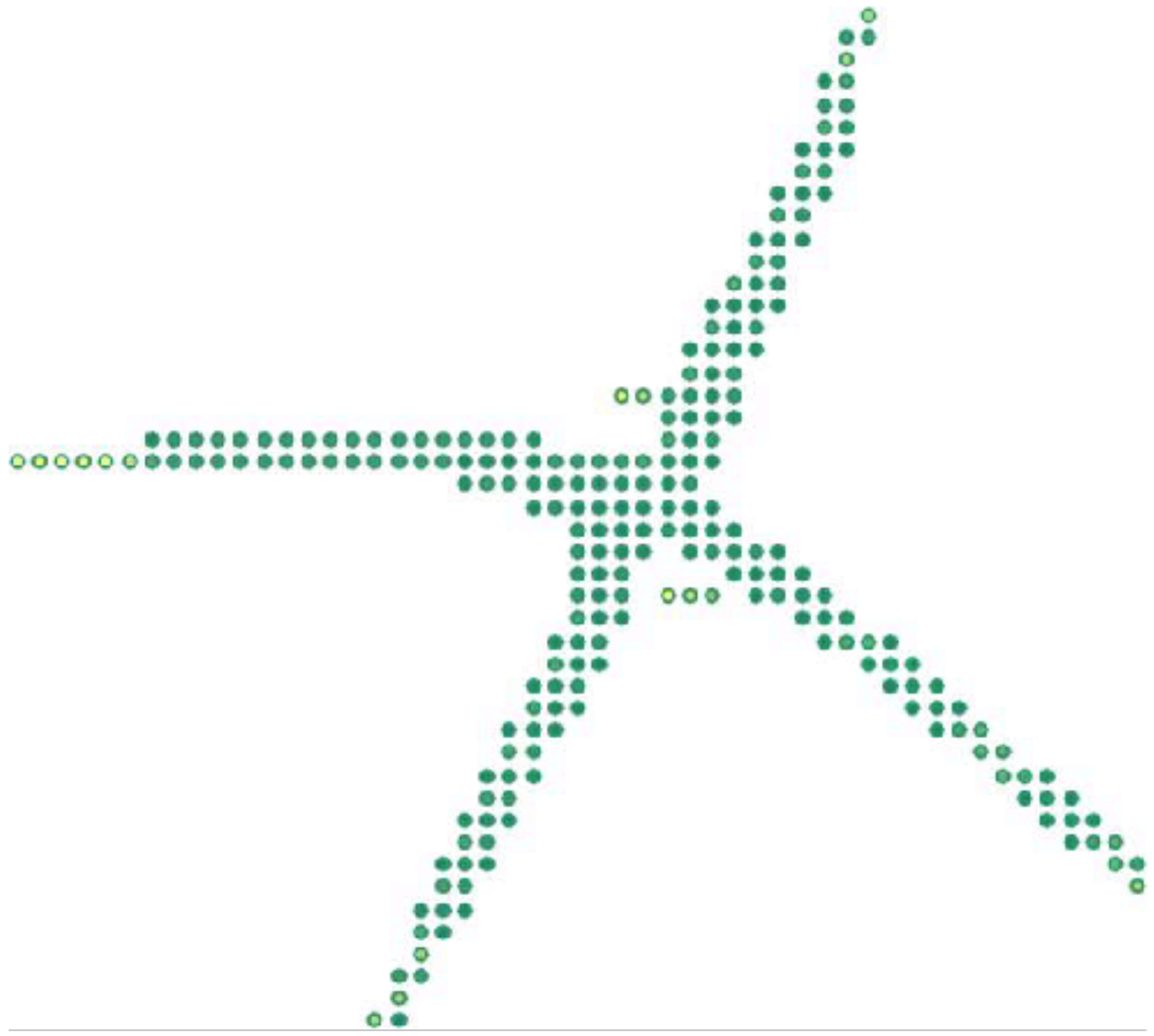


initialization: skeleton points X
uniformly sampled from Q

From intuition to a solution

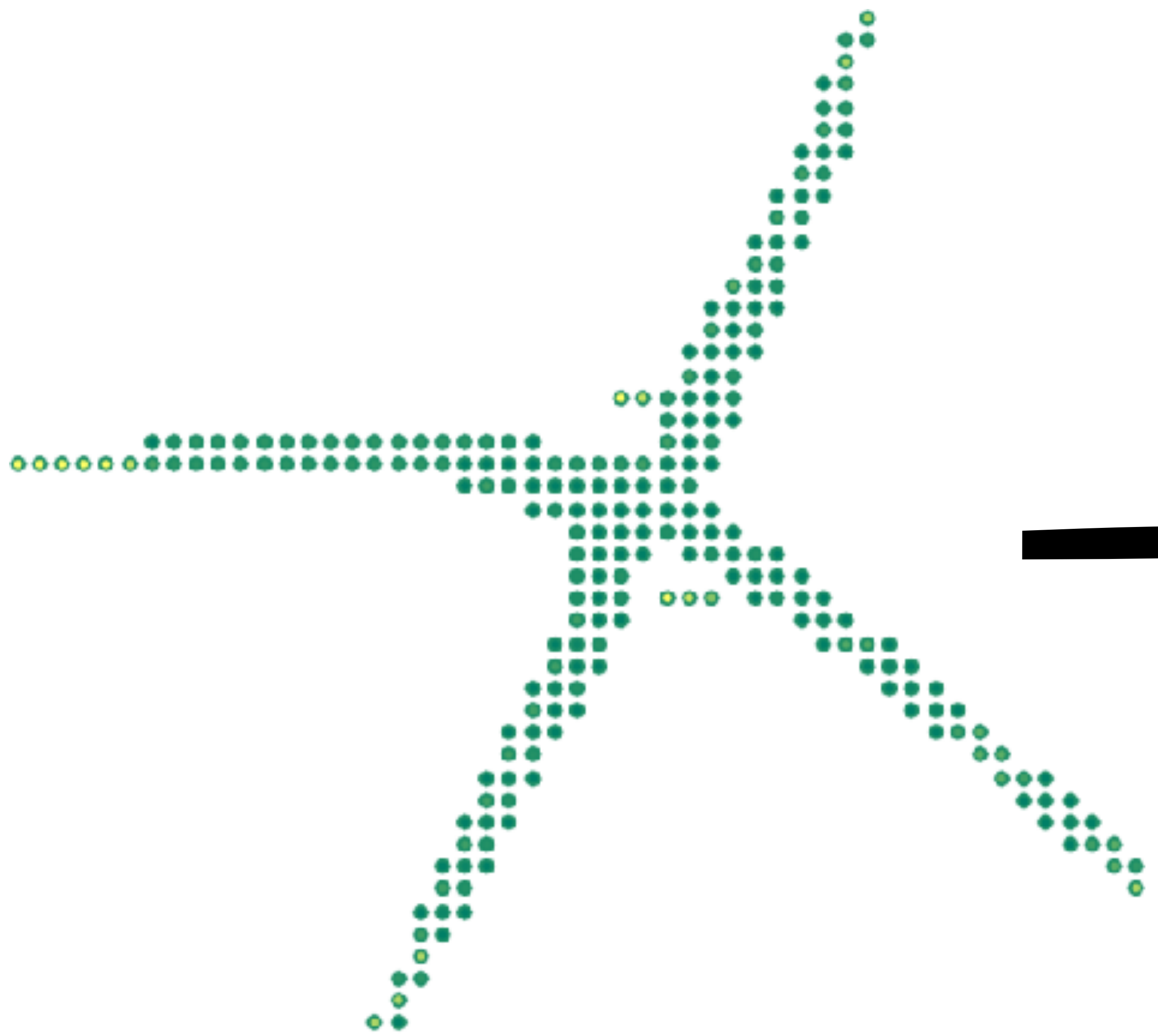


initialization: skeleton points X
uniformly sampled from Q

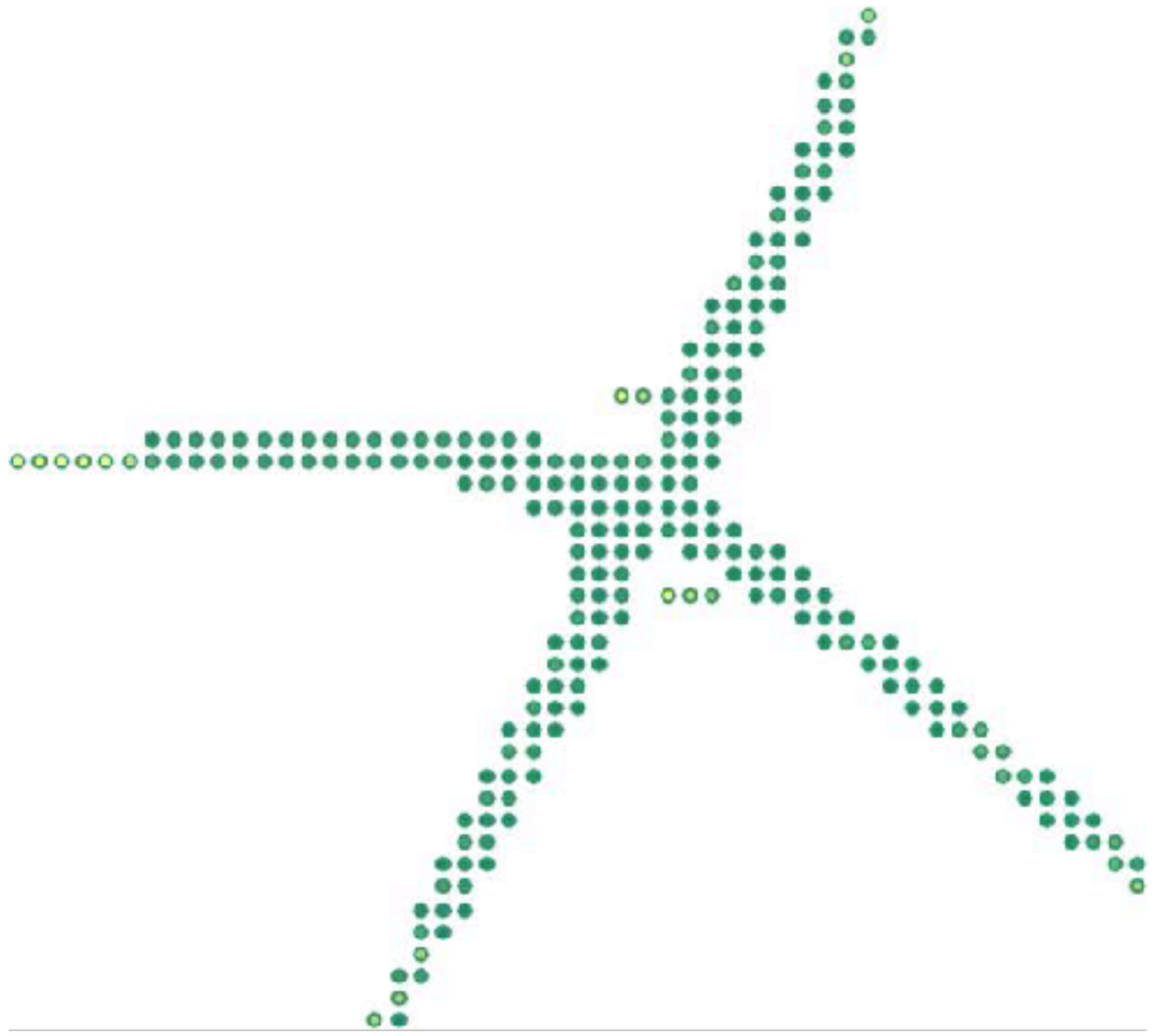


extract branches

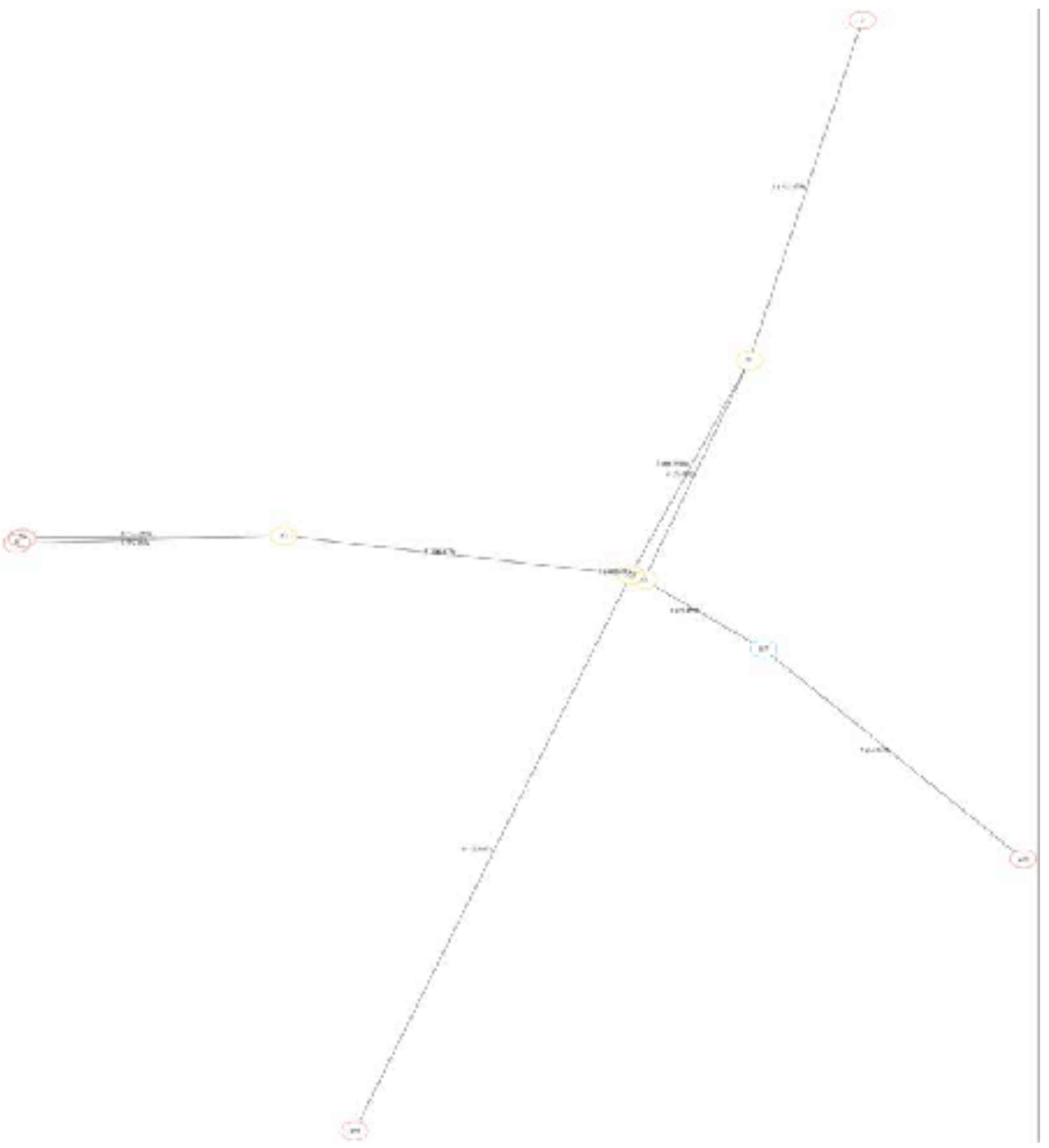
From intuition to a solution



initialization: skeleton points X
uniformly sampled from Q

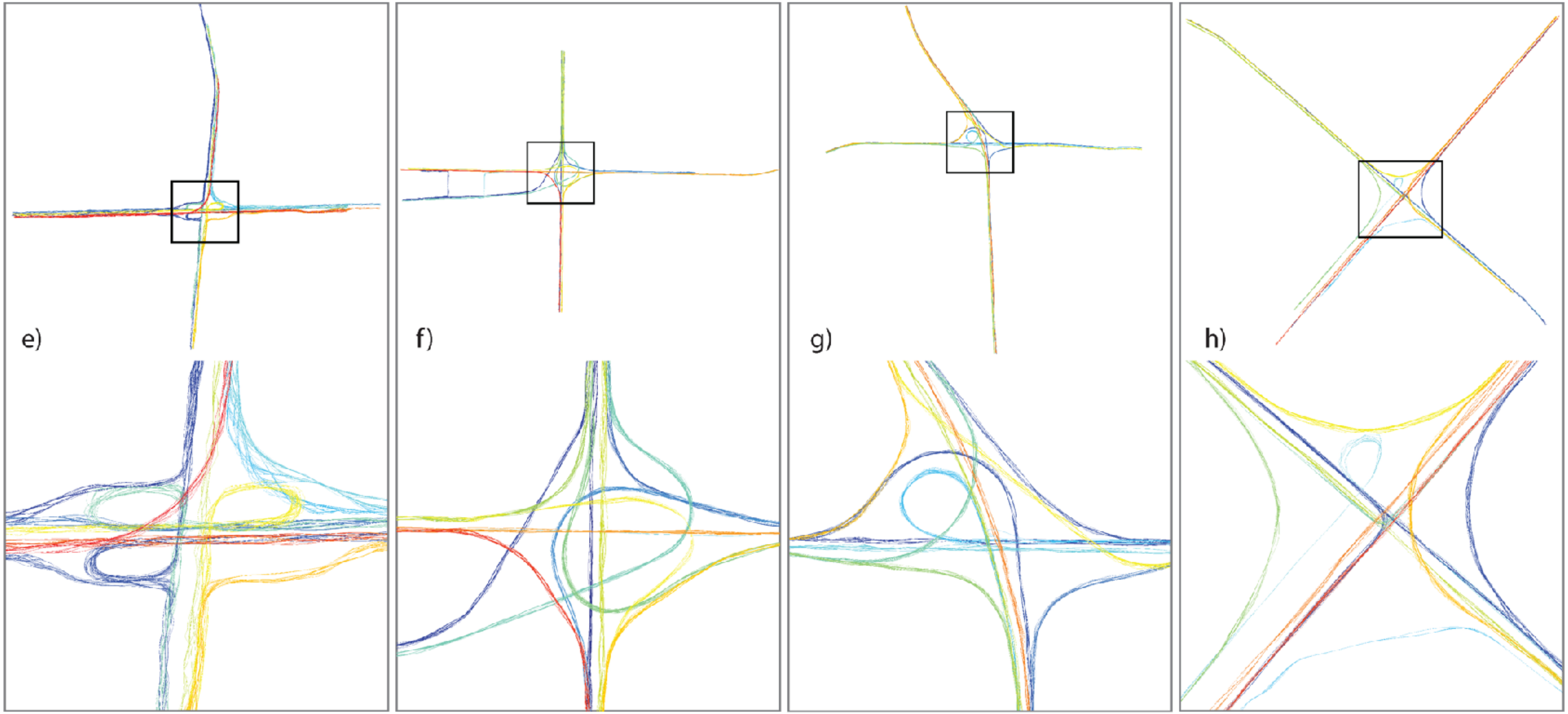


extract branches



compute skeleton

Benchmark Results (BM_1-BM_3)



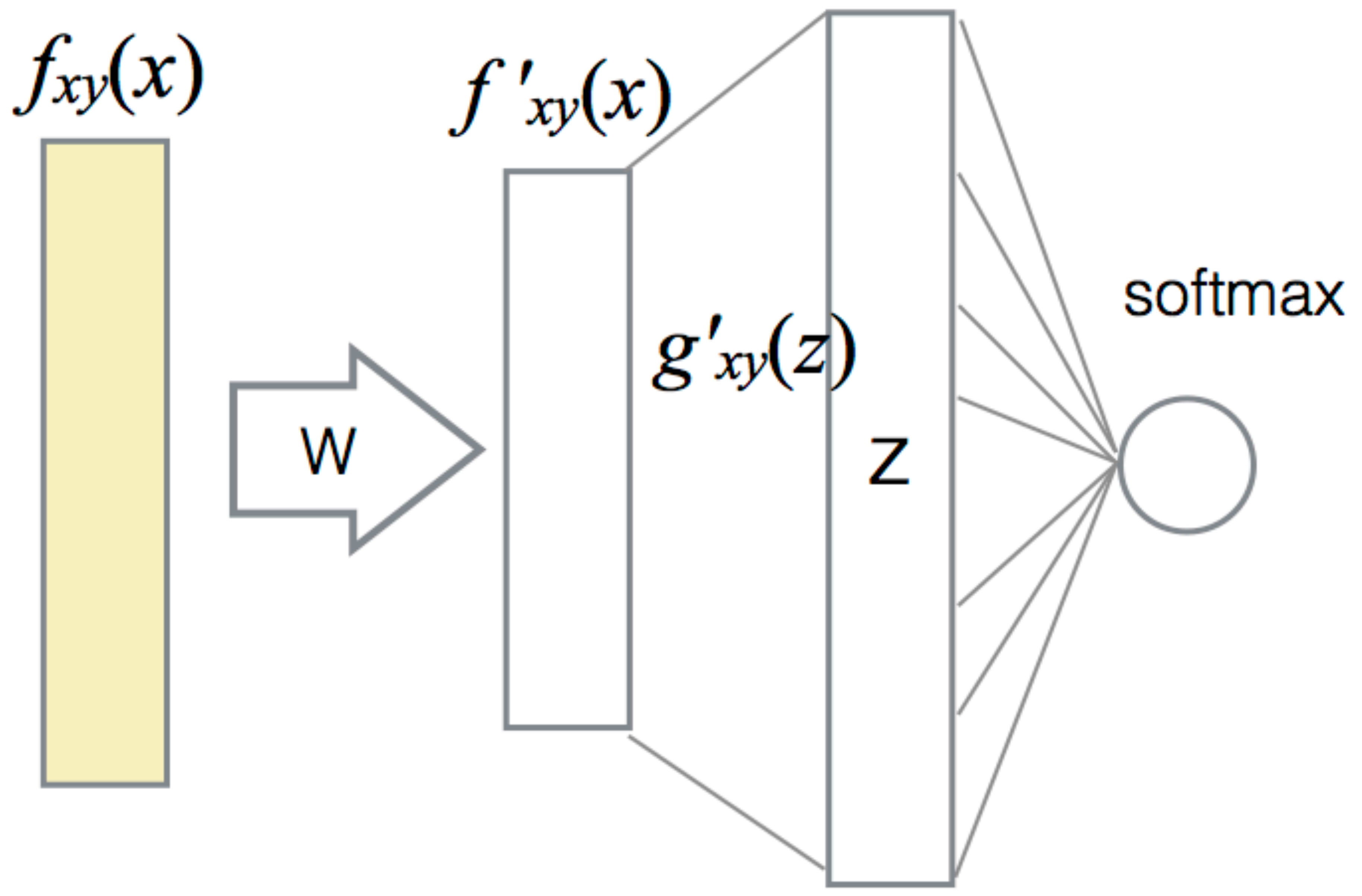
sampling interval: 20 sec

running time: 1-2 min

2. Talk to other people

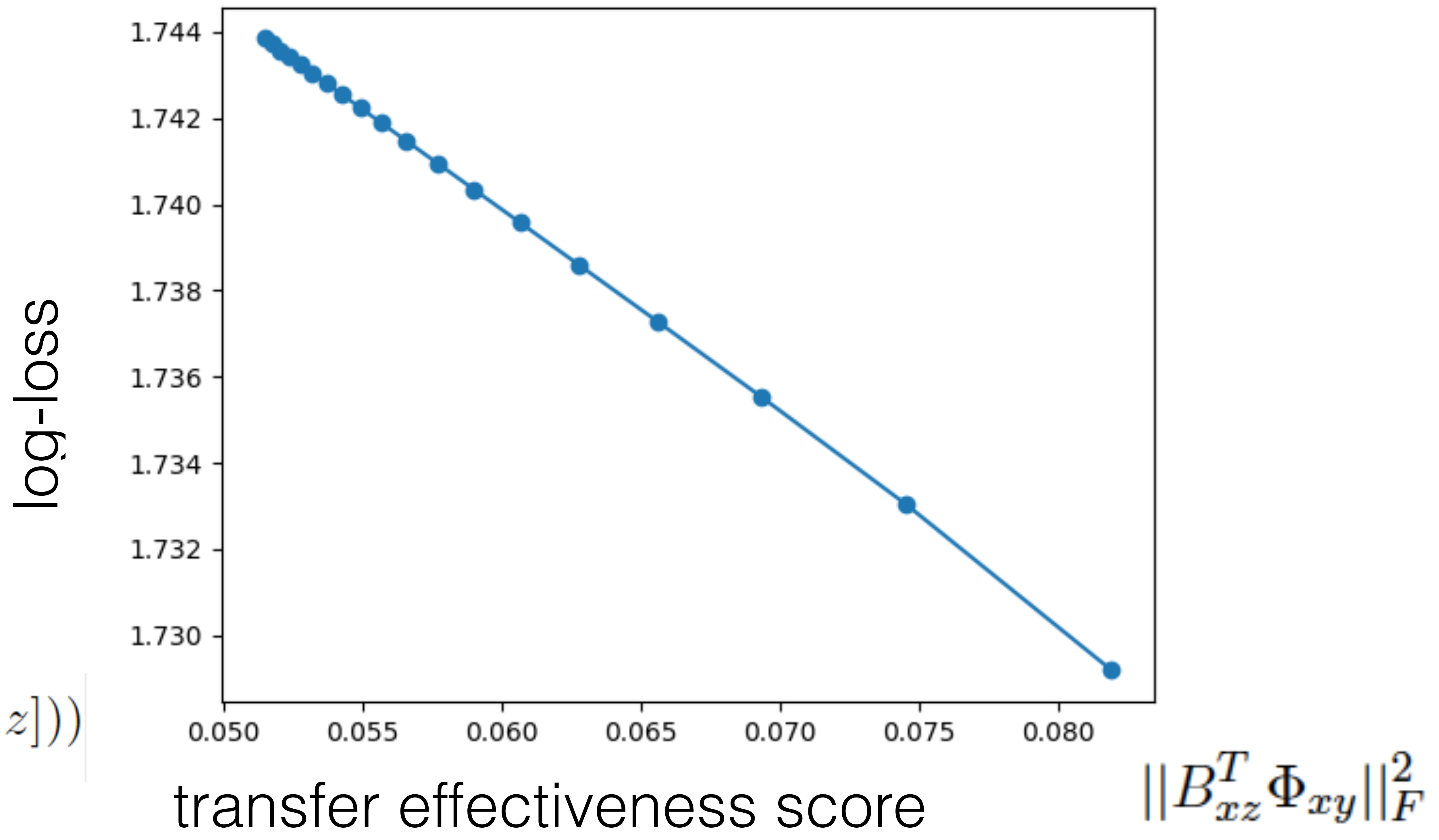
3. Don’t get stuck on a side goal

Example: Our first transferability project



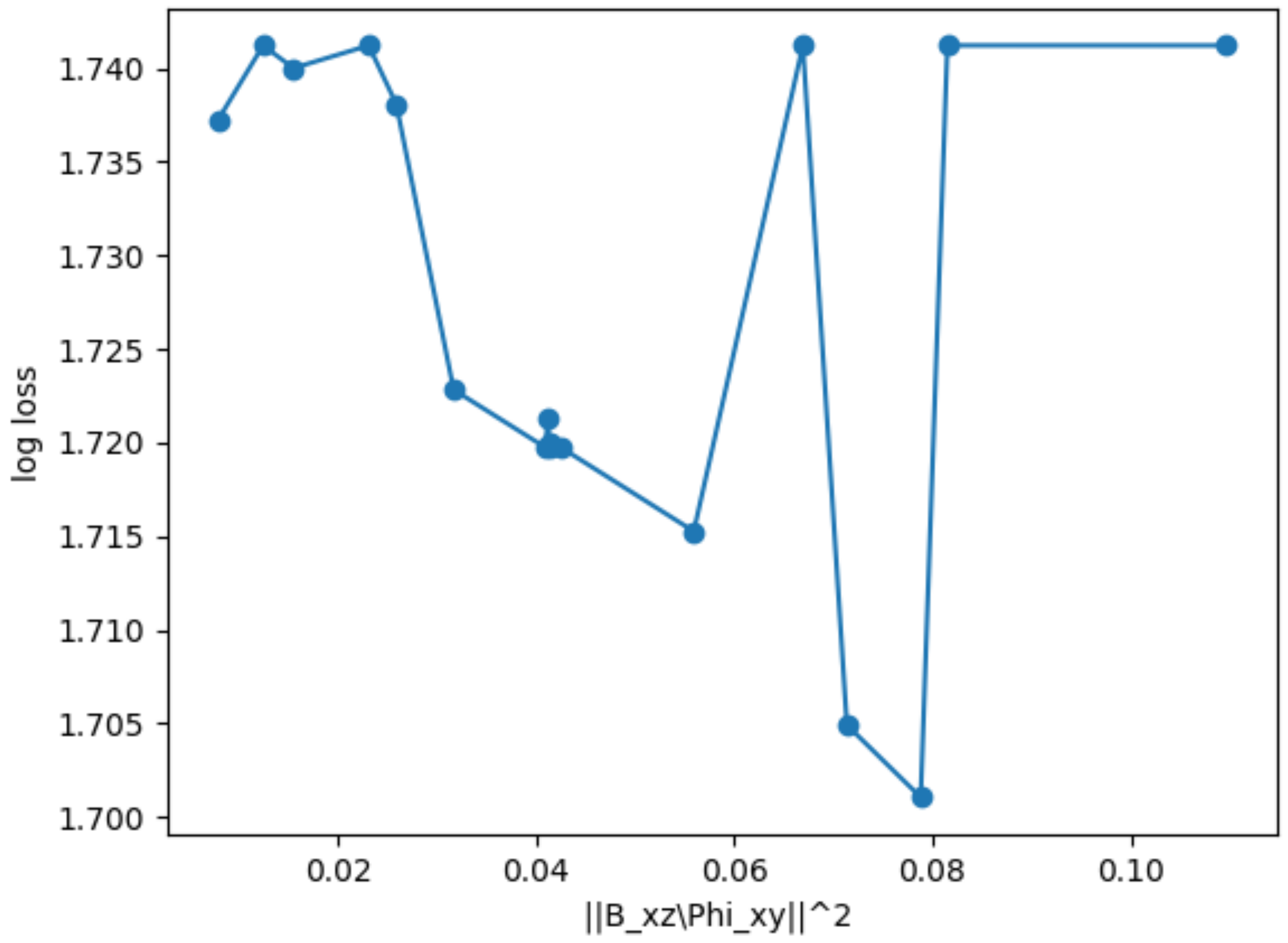
$$||B_{xz}^T \Phi_{xy}(\Phi_{xy}^T \Phi_{xy})^{-\frac{1}{2}}||_F^2 = tr(cov(f_{xy}(X))^{-1} cov(\mathbb{E}[f_{xy}(X)|Z=z]))$$

Theoretical Result: Perfect



Empirical result: Very Poor

2) Use neural-network implementation

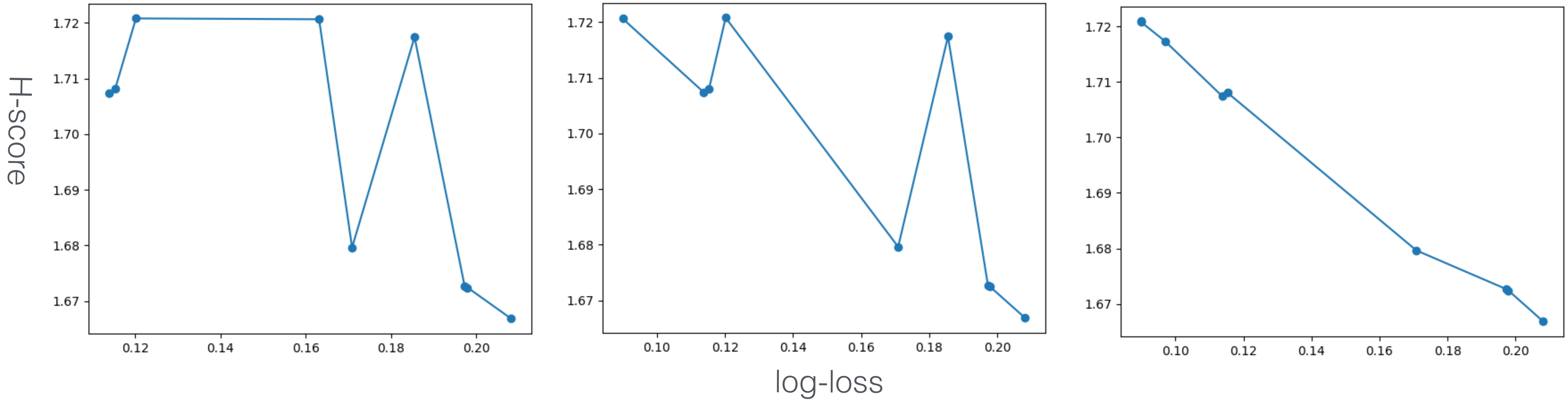


Bring theory to practice

Previous issues on computing ACE features f_{xy}

- local minimum (**train multiple trails**)
- cov(f) is near singular (**use pseudoinverse with larger tolerance threshold**)

tolerance=1e-6 tolerance=1e-5 tolerance=1e-4

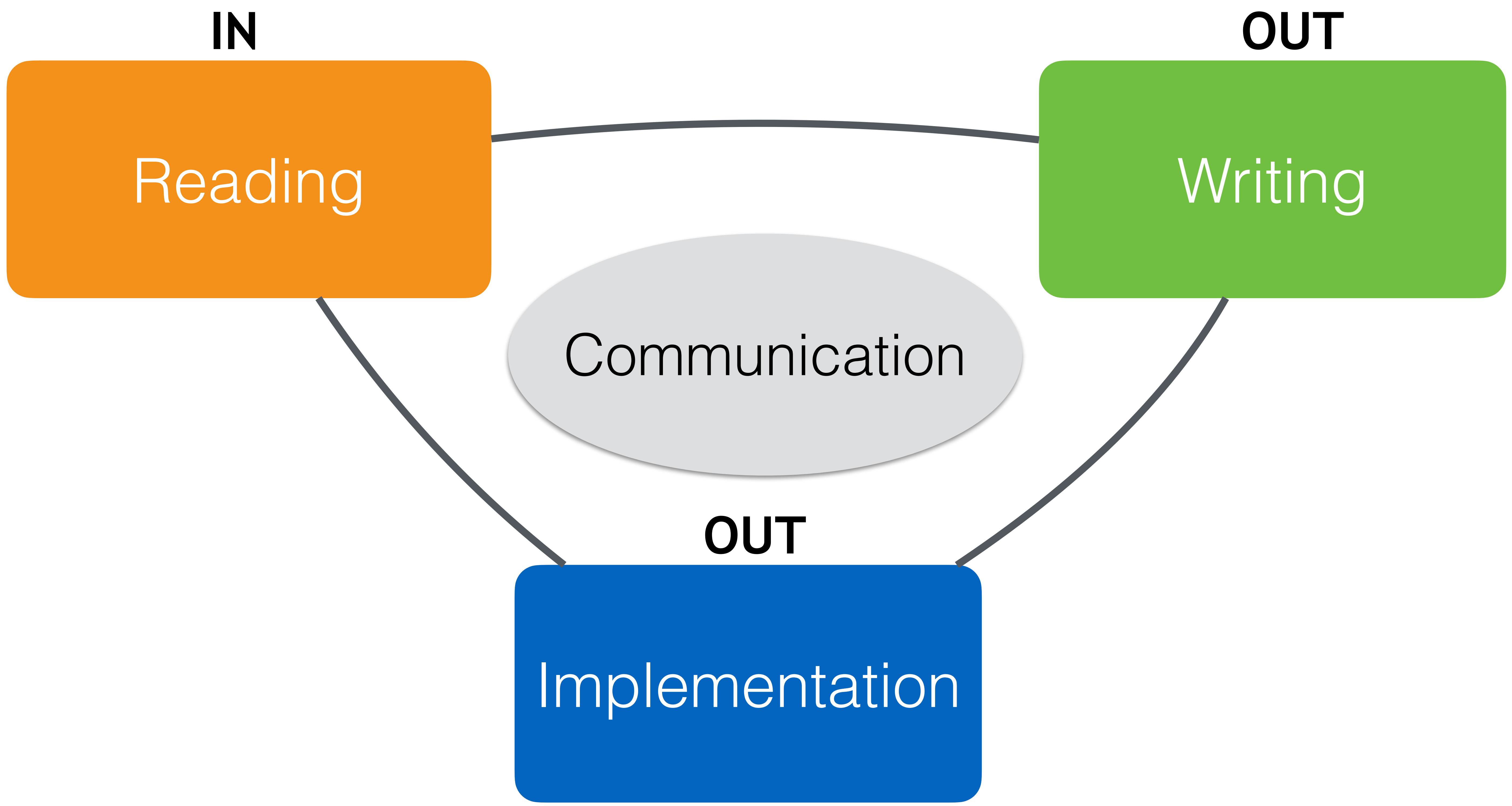


4. Trust in the theory

1. Aim high, trust your intuition
2. Talk to other people
3. Don’t get stuck on a side goal
4. Trust in the theory

Essential Skills for EECS Students

besides technical knowledge

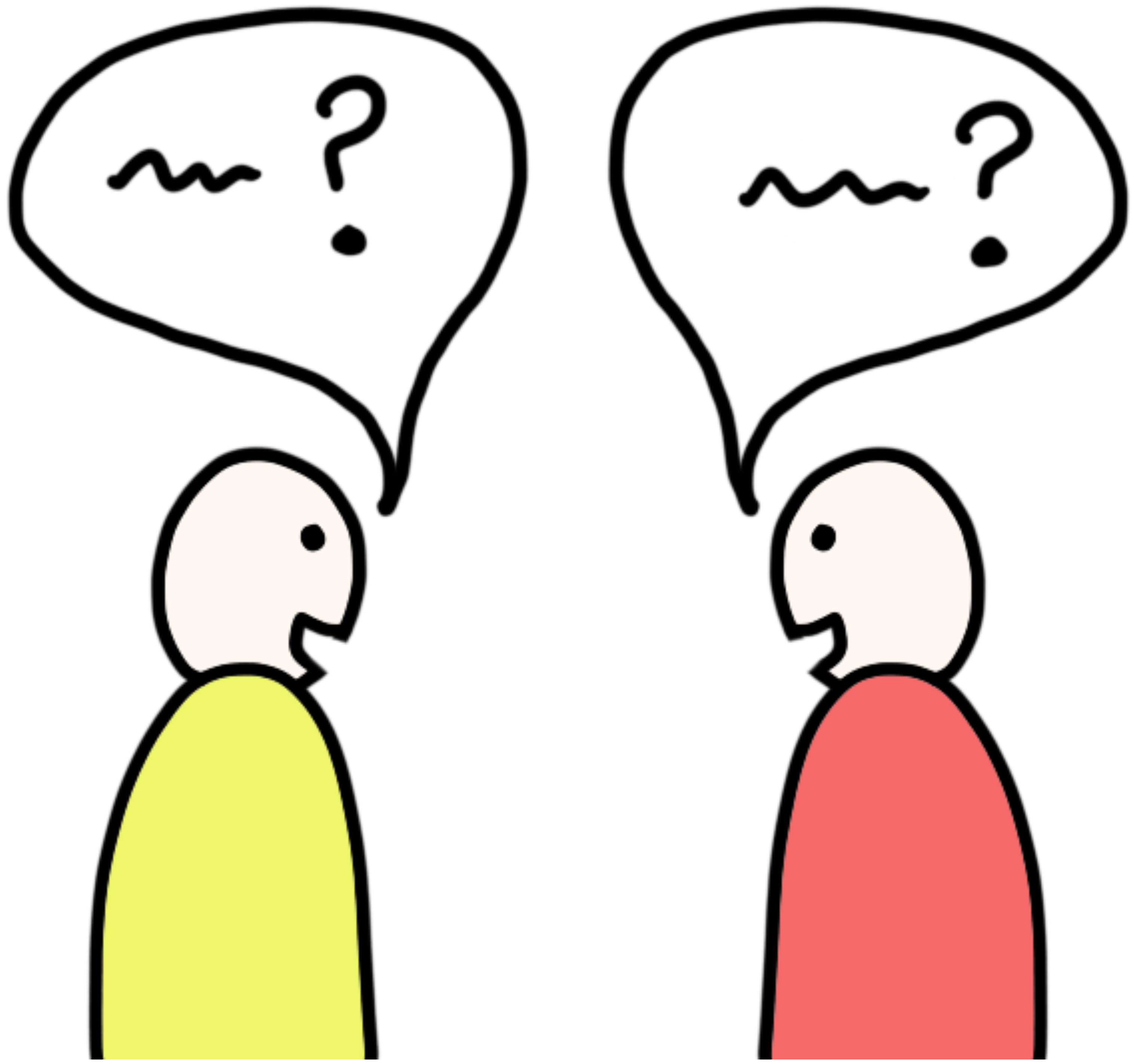


How to Read

- Take notes on reading
- The three-pass approach (http://ccr.sigcomm.org/online/files/p83-keshavA.pdf)
-
    - First pass: 5-10 min
    - Second pass: 1 hour
    - Third pass: 1-5 hours

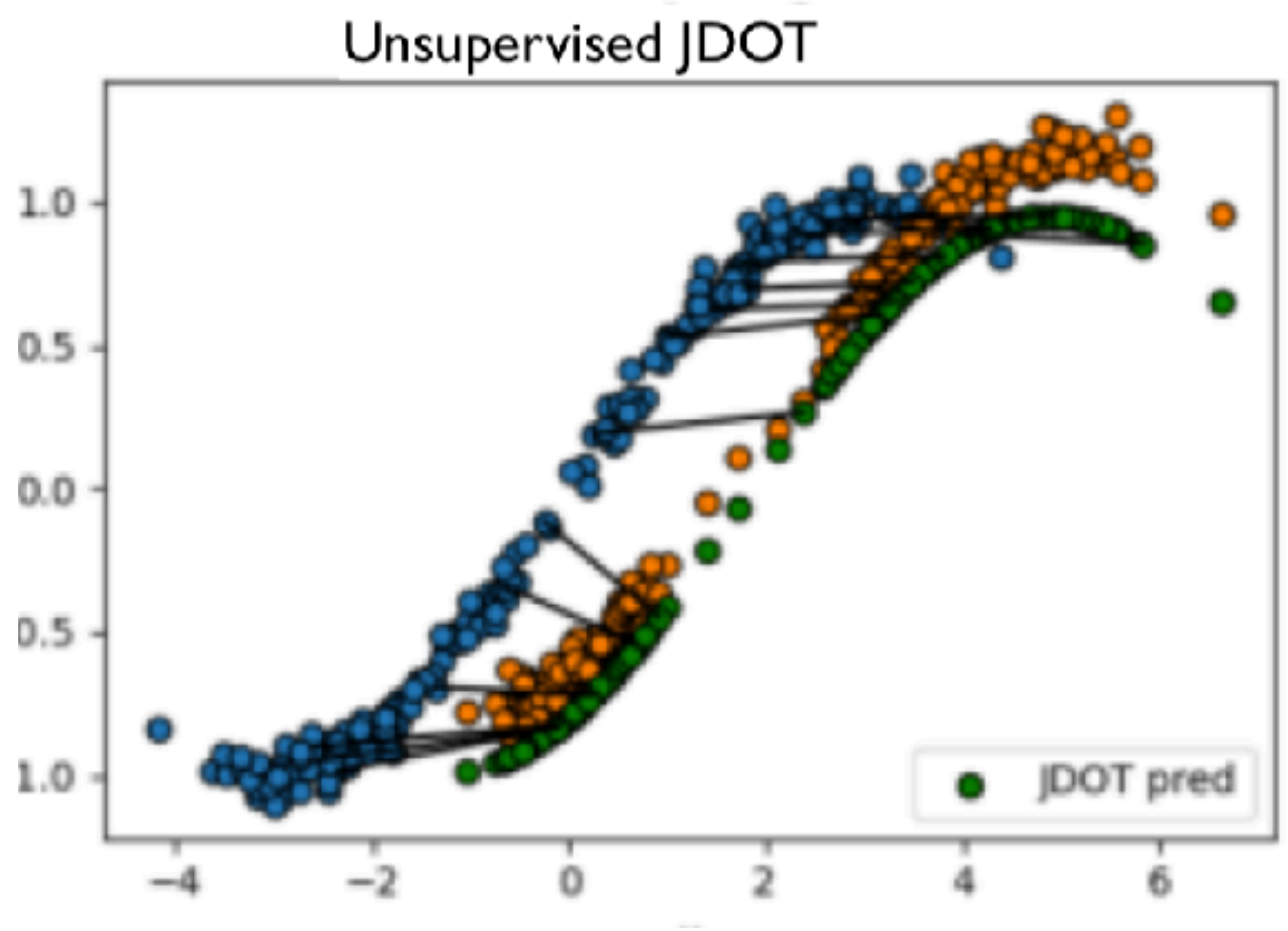
Test Your Understanding

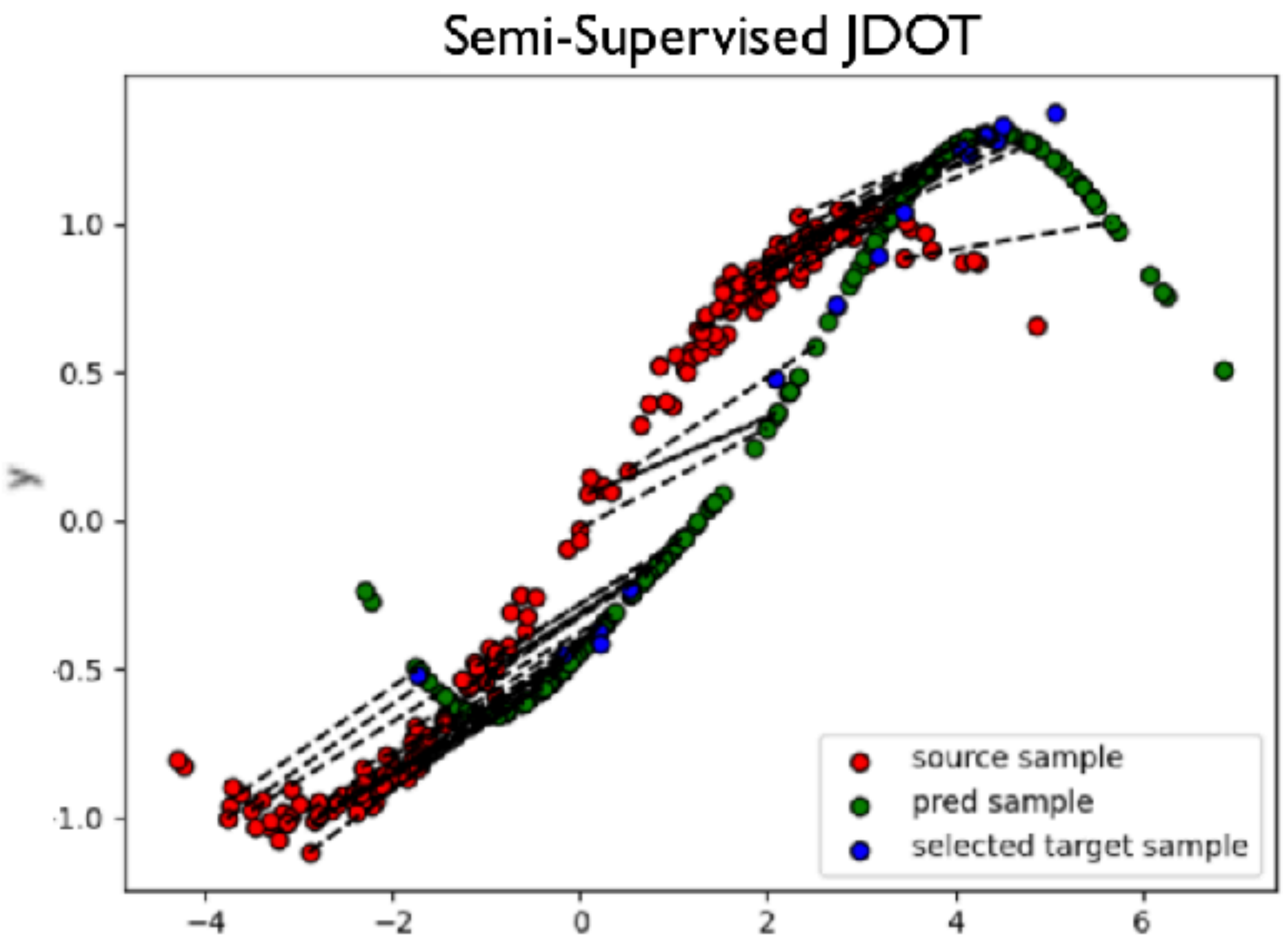
- Explain to someone else
  - Labmates
  - Friends & family



Test Your Understanding

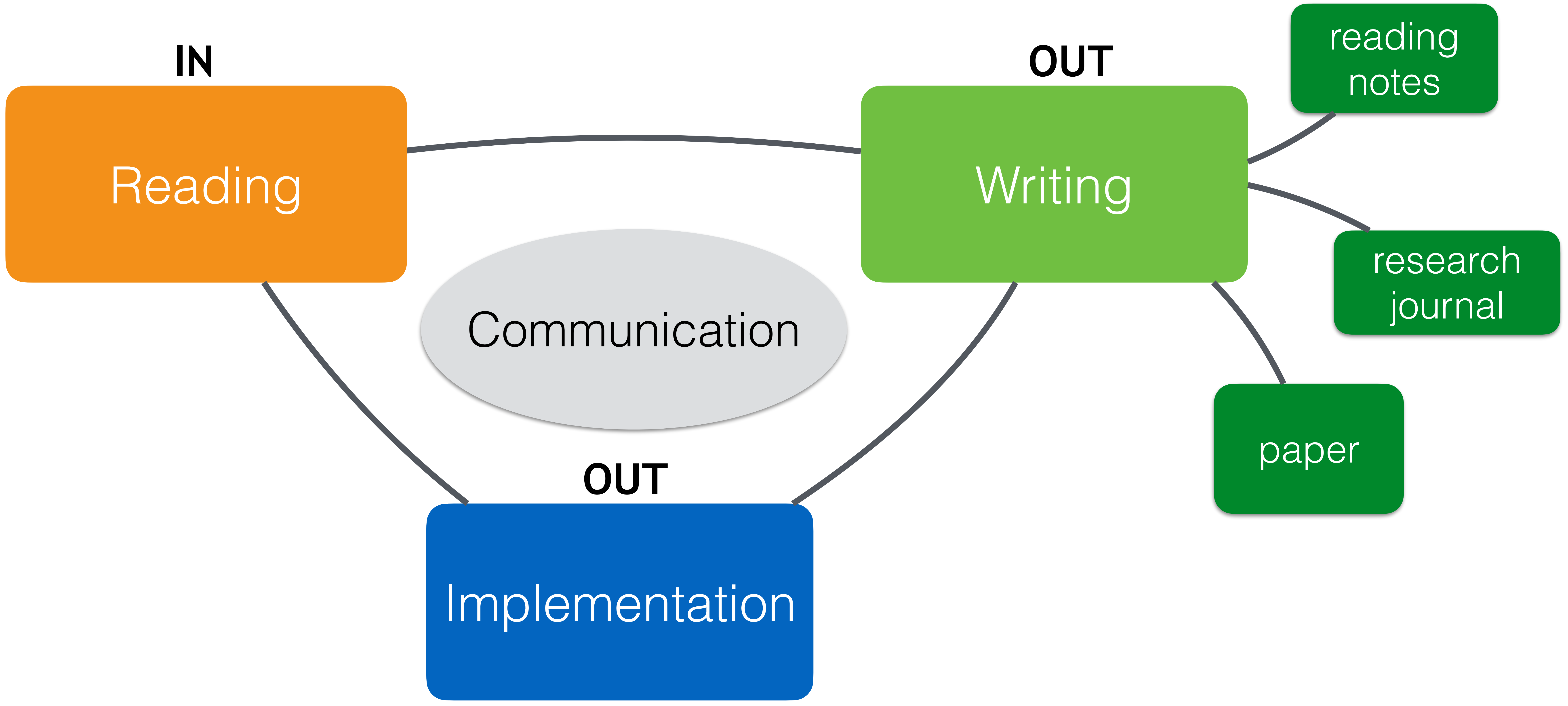
- Reimplement it
  - Make toy examples
  - Break complex complem apart





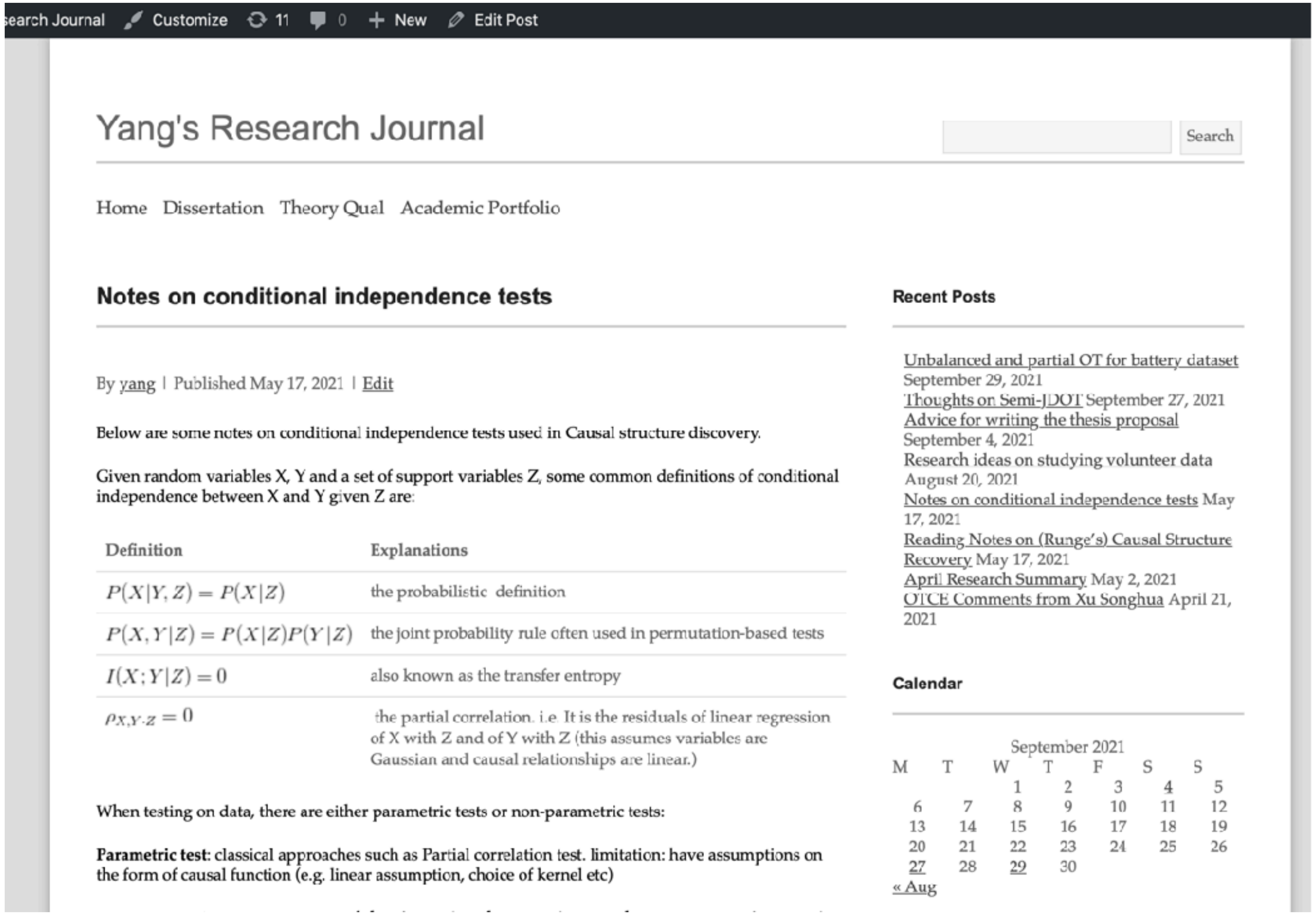
Essential Skills for Lab2C (EECS) Students

besides technical knowledge



Research Journal

- What to write?
  - reading summaries
  - research progress and ideas
  - reflection of meeting


earch Journal Customize 11 0 + New Edit Post

Yang's Research Journal

Search

Home Dissertation Theory Qual Academic Portfolio

Notes on conditional independence tests

By yang | Published May 17, 2021 | Edit

Below are some notes on conditional independence tests used in Causal structure discovery.

Given random variables X, Y and a set of support variables Z, some common definitions of conditional independence between X and Y given Z are:

Definition	Explanations
$P(X\|Y,Z) = P(X\|Z)$	the probabilistic definition
$P(X,Y\|Z) = P(X\|Z)P(Y\|Z)$	the joint probability rule often used in permutation-based tests
$I(X;Y\|Z) = 0$	also known as the transfer entropy
$\rho_{X,Y \cdot Z} = 0$	the partial correlation. i.e. It is the residuals of linear regression of X with Z and of Y with Z (this assumes variables are Gaussian and causal relationships are linear.)

When testing on data, there are either parametric tests or non-parametric tests:

Parametric test: classical approaches such as Partial correlation test. limitation: have assumptions on the form of causal function (e.g. linear assumption, choice of kernel etc)

Recent Posts

Unbalanced and partial OT for battery dataset September 29, 2021
Thoughts on Semi-JDOT September 27, 2021
Advice for writing the thesis proposal September 4, 2021
Research ideas on studying volunteer data August 20, 2021
Notes on conditional independence tests May 17, 2021
Reading Notes on (Runge's) Causal Structure Recovery May 17, 2021
April Research Summary May 2, 2021
OTCE Comments from Xu Songhua April 21, 2021

Calendar

September 2021

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

« Aug

Other Formats

Research Progress Report (Week 11-12)

Yang Li

July 2, 2012

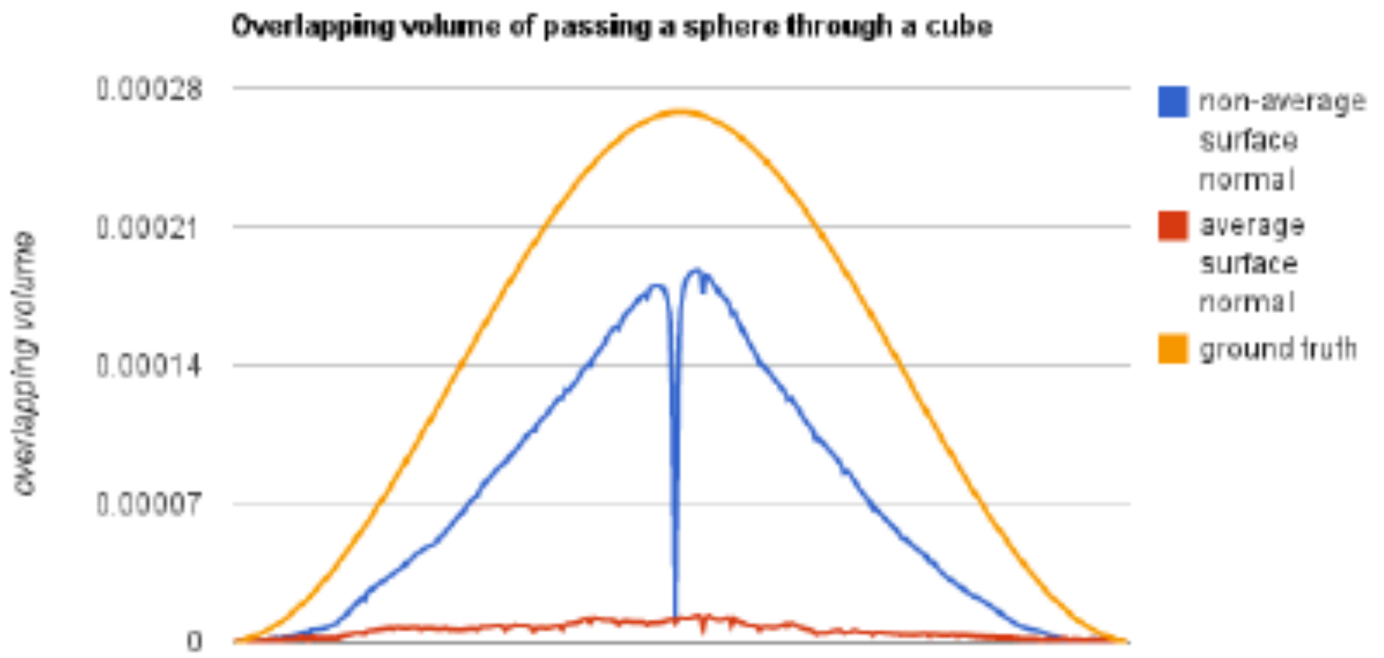
The main task for the past two weeks has been debugging the volume query and testing the collision response force. I have made large improvement on computing the overlapping volume. However, the direction of the contact normal force computed in the current implementation is still incorrect.

1 Improve volume query

1.1 Resolve implementation issues

One problem with the previous implementation is the size of the hierarchical spheres. After carefully reviewing the literature on finding the minimum enclosing sphere of a collection of spheres. I found that the previous method that builds upon the minimum sphere of sphere centers is incorrect. I studied two alternative algorithms, one is a better iterative method, though it do not guarantee the optimal result. The other is an optimization-based method that yields the exact solution [1]. I implemented the former since the latter requires solving an non-linear optimization problem.

The major problem causing the problem of underestimated volume is in fact a bug in the implementation. i.e. When the intersecting volume between 2 innerspheres is 0 (no collision), an extra step is needed to restore the cumulative volume computed in the previous step. By resolving this issue, I obtained a volume curve much closer to the groud truth. See the blue curve in Figure 1



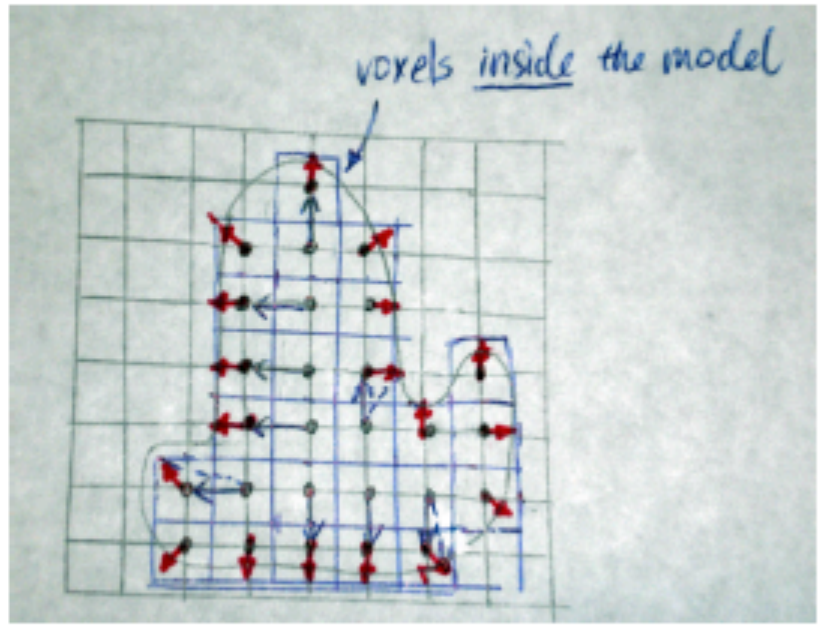


Figure 1: Distance map in 2D. Voxels are outline in blue, with centers shown as black dots. (The voxel centers are defined as corners within the surface in the original grid.) Distance vectors of the boundary points are drawn as red arrows; that of the non-boundary points are drawn in blue.

- **Binary sampled data (voxels).** A common approximation method is analogous to the matching cube algorithm. i.e. After computing the triangle faces, the distance to the nearest face is computed.Gibson wrote a survey on existing methods of generating distance map from sampled data [1]. Besides exhaustive search using distance metric and the marching cubes, Gibson also discussed the central point method, i.e. first construct the surface by connects points at the center of each boundary cube, then compute the distance map from the mesh of center points using Euclidean distances. This simple method, however, beats all others in the 2D experiment shown in the paper.

Computing distance maps in Parapluie

Since our project already has conversion between surface, volume and voxel data implemented. We could use any distance field computation method. However, as our project mainly uses medical input, it is more natural to directly approximate the distance map from binary sampled points (e.g. voxels), even assuming the mesh is not available. Therefore, I came up with the following basic algorithm for 2D:

Input Point set P containing all voxel centers

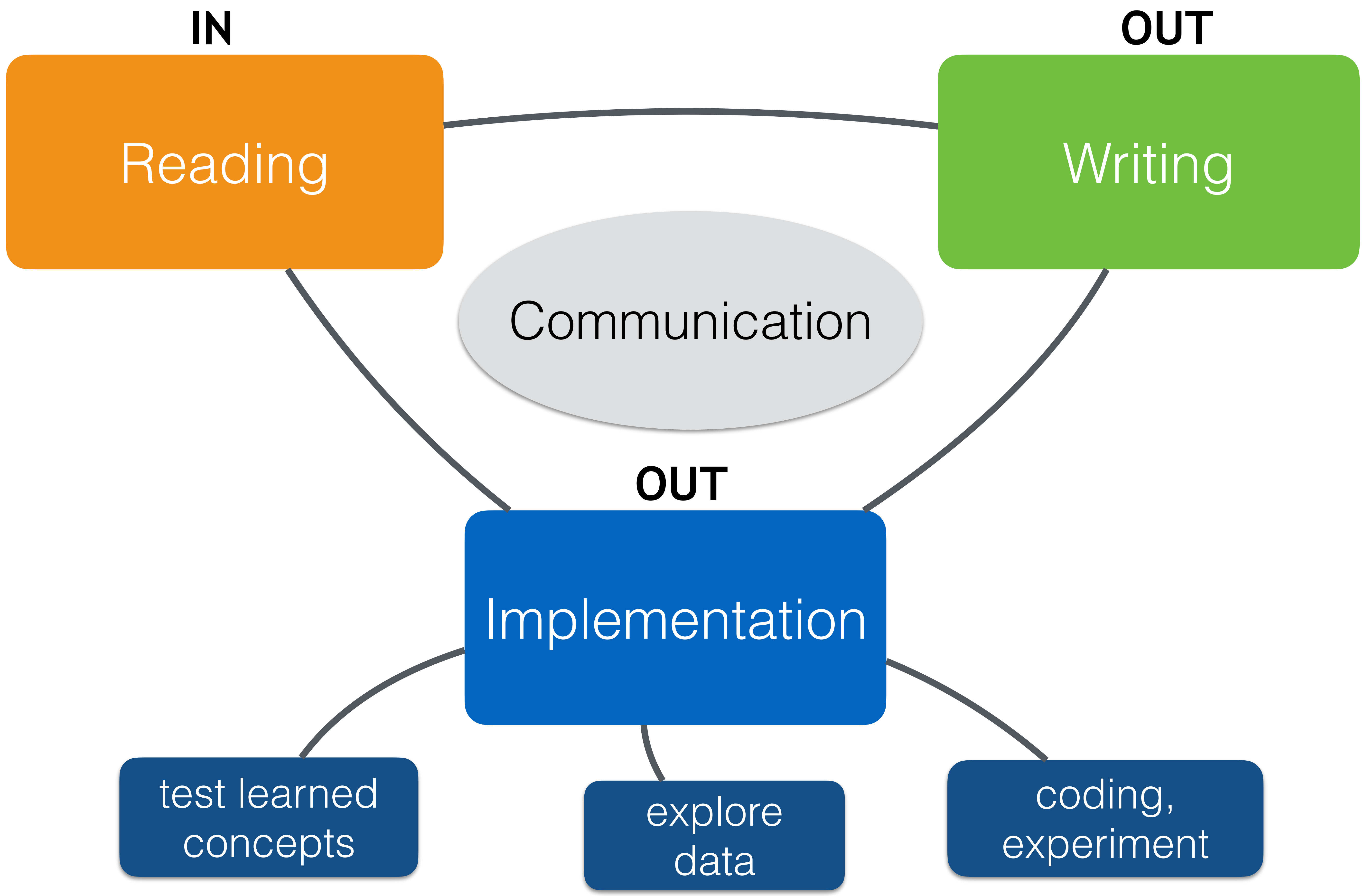
Output Distance map D that maps point p with index (i, j) to d_p, a vector from $P[i][j]$ to the (approximated) closest point on a nearest surface.

Algorithm outline

1. Find all boundary voxels $B \subset P$ and estimate their normal vectors. For each point

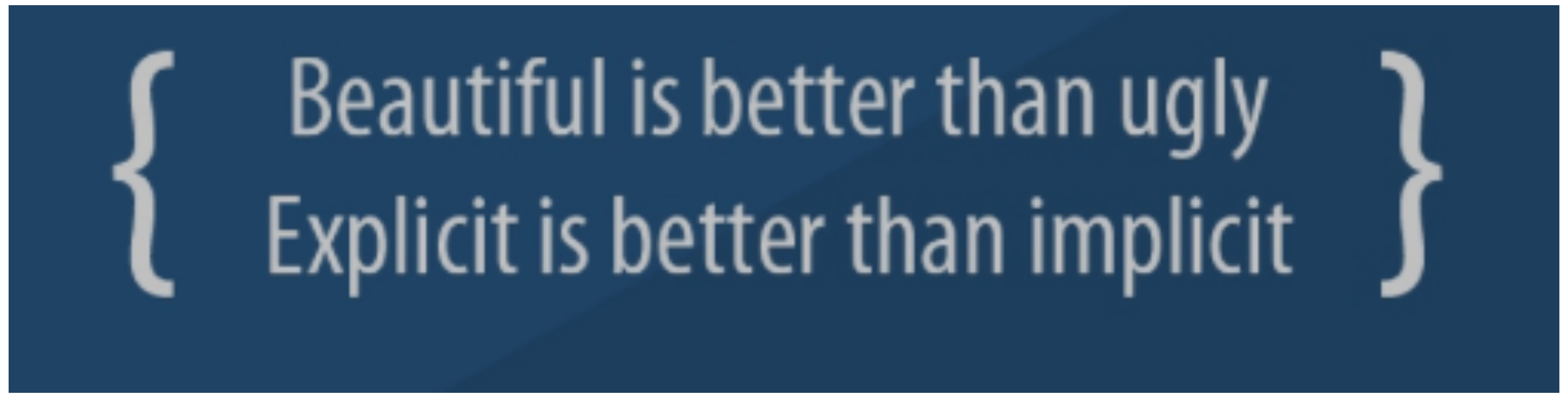
Essential Skills for Lab2C (EECS) Students

besides technical knowledge



Coding

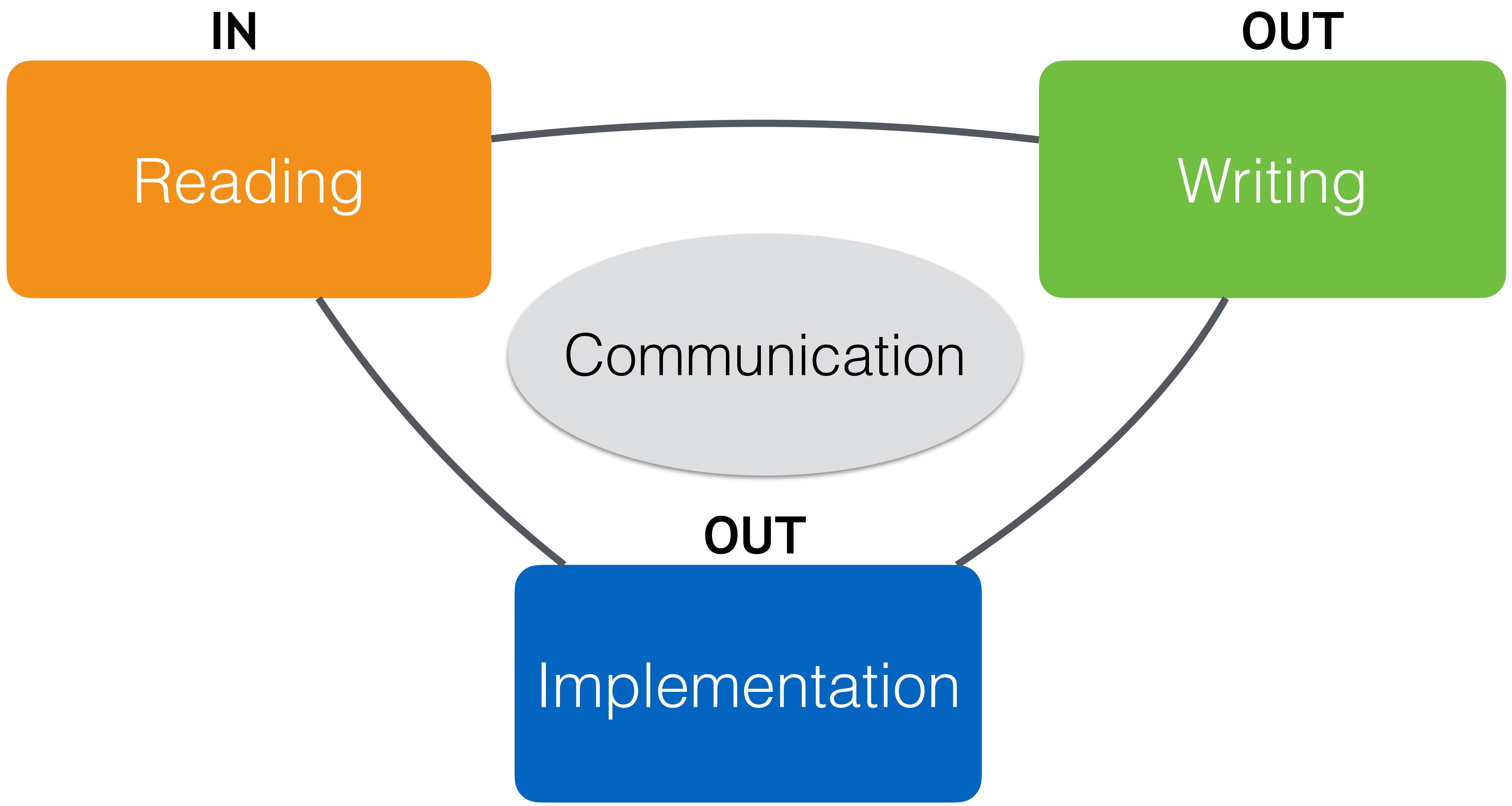
- Back up your code (and everything you produce) !!!
- Use version control: git
  - see：http://10.8.6.22/wiki/index.php/File:GITHUB.pdf
- Develop an asthetic for good coding and documentation
  - object oriented vs precedual vs functional programming



from Zen of Python https://www.python.org/dev/peps/pep-0020/

Essential Skills for EECS Students

besides technical knowledge



Communication — Presentation

- Weekly research meeting (progress update)
  - before the meeting: prepare an agenda (bring notes or visual aid)
  - after the meeting: summarize and make plans
- Group meeting/conference talks
- English or Chinese？

Summary

- Make a weekly worksheet to keep track of your research acitivity in all three fields
    - Beaware of concentrating too much on one field only
- Identify your strength and weakness, and practice

Links

- Lab-hosted overleaf: http://10.8.6.22:8031
- Lab-hosted gitlab: http://10.8.6.22:88
- Lab2c wiki: http://10.8.6.22/wiki/index.php
- yang's group wiki: http://yangli-feasibility.com/wiki/
- Good reads:
  - You & Your Research, Richard Hamming.
  - The Art of Ding Science and Engineering, Richard Hamming.
  - The Bus Ticket Theory of Genius, Paul Graham