# Uses of Persistence for Interpreting Coarse Instructions

ICRA Workshop on Emerging Topological Techniques in Robotics
20th May 2016


**Subramanian Ramamoorthy**

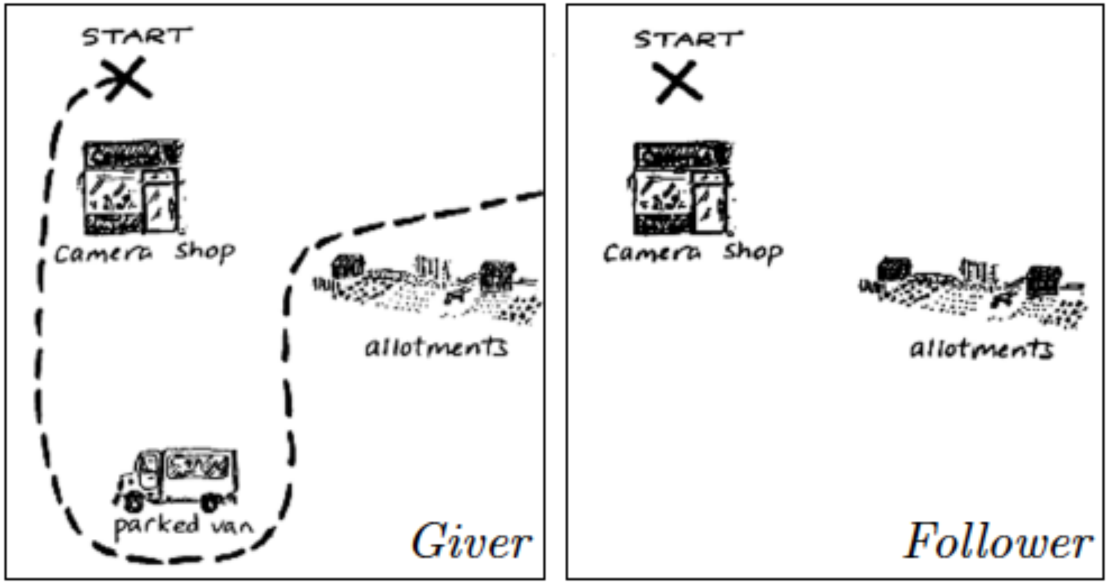
School of Informatics

The University of Edinburgh


Rad | Robust Autonomy & Decisions

# Scenario: Human-Robot Co-Work

# How Do People Give Instructions for Spatial Navigation?



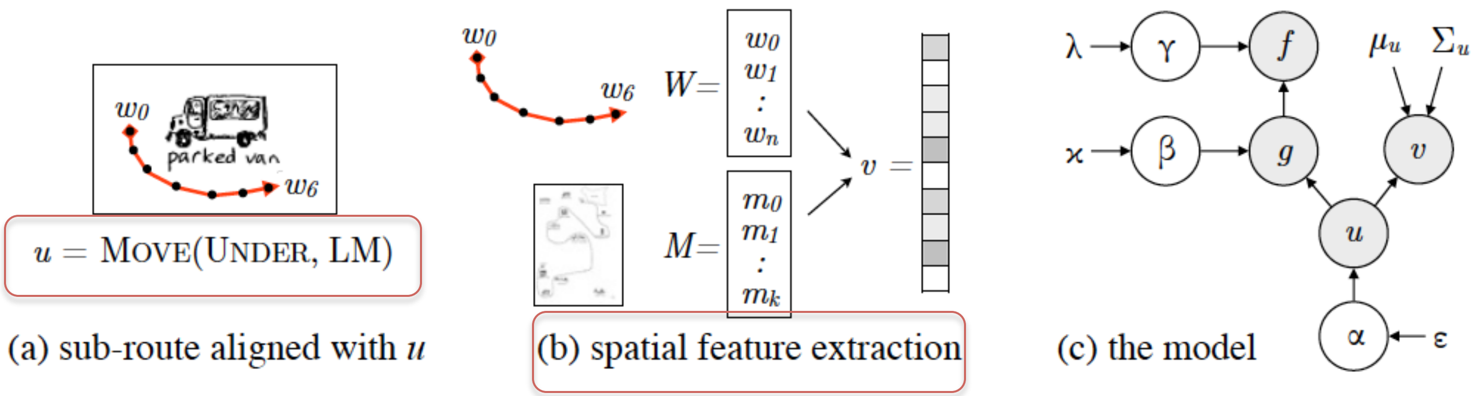

**HCRC Map Task**: Instruction Giver's task is to communicate a route to a Follower, whose map may differ. Route is Giver's goal which the Follower tries to infer.

A. Eshky, B. Allison, S. Ramamoorthy, M. Steedman, **A generative model for user simulation in a spatial navigation domain**, In Proc. EACL 2014.

# Example Data for Map Task

| **Natural Language** | **Semantic Representation** |
|---|---|
| *G*: you are above the camera shop | *Instruct* POSITION(ABOVE, LM) |
| *F*: yeah | *Acknowledge* |
| *G*: go left jus– just to the side of the paper, ⋆<br>then south,<br>under the parked van ◇<br>you have a parked van? | *Instruct* MOVE(TO, PAGE_LEFT) ⋆<br>*Instruct* MOVE(TOWARDS, ABSOLUTE_SOUTH)<br>*Instruct* MOVE(UNDER, LM) ◇<br>*Query-yn* |
| *F*: a parked van no | *Reply-n* |
| *G*: you go– you just go west, ⋆<br>and down,<br>and then you go along to the– you go east ◇ | *Clarify* MOVE(TOWARDS, ABSOLUTE_WEST) ⋆<br>*Clarify* MOVE(TOWARDS, ABSOLUTE_SOUTH)<br>*Clarify* MOVE(TOWARDS, ABSOLUTE_EAST) ◇ |
| *F*: south then east | *Check* |
| *G*: yeah | *Reply-y* |

# Generative Model: Behaviour in Map Task



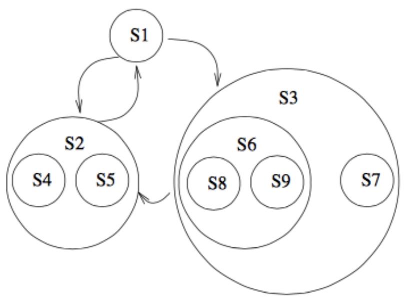

(a) sub-route aligned with $u$ (b) spatial feature extraction (c) the model

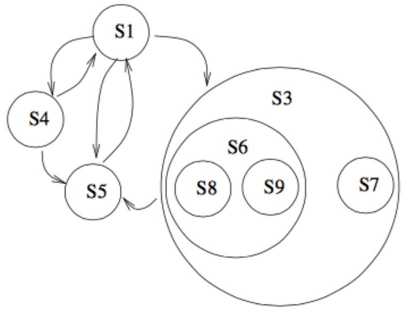
[A. Eshky et al., EACL 2014]

**Core Question: Can we have multi-scale state estimation to deal with evidence from multiple such modalities, of varying coarseness?**

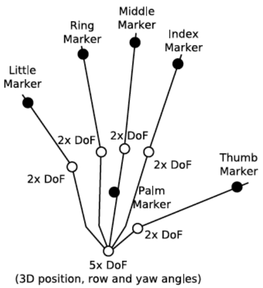
# Representative Prior Work

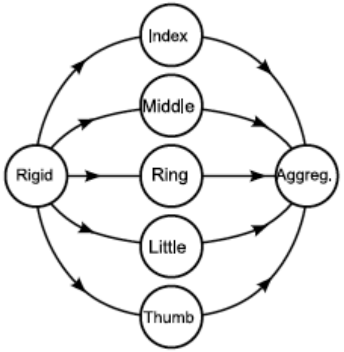
- Variable resolution particle filtering [Verma, Thrun, Simmons IJCAI '03 ]: clump states to define 'macrostate'



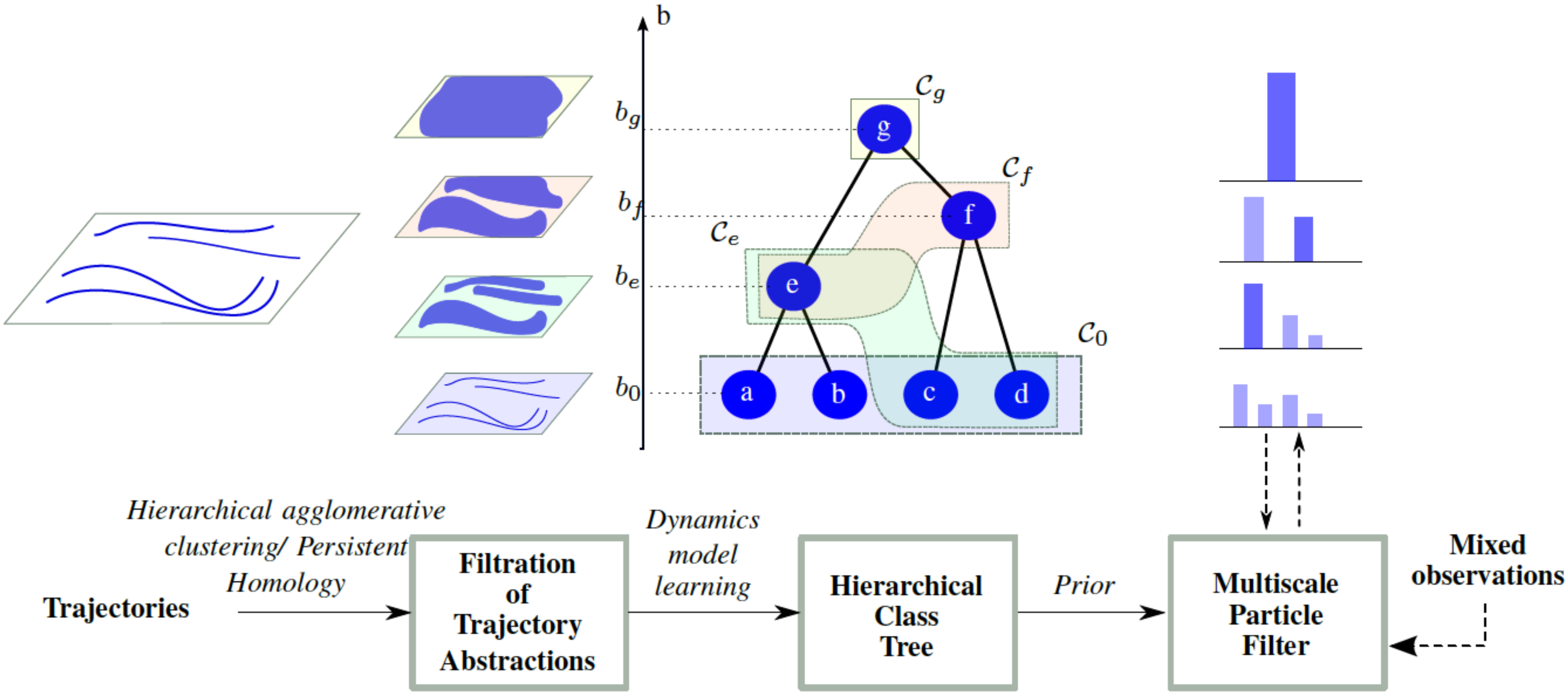




- Hierarchical subspace filtering [e.g., Brandao et al. '06]: track subset of variables separately and aggregate



# Our Approach




[M. Hawasly, F.T. Pokorny, S. Ramamoorthy, **Hierarchical Filtering with Spatial Abstractions**, Manuscript in preparation]

# Hierarchical Agglomerative Clustering

- Iterated operation: merge two clusters at lower level to get a single new cluster
- Yields a tree data structure
  - leaves are the individual data items
  - root node is the cluster made by merging all data points
- Order of merging depends on distance between clusters, such that the pair with the smallest distance is merged first.
- Every new cluster can be assigned a distance value at which it gets created.

# Simple Clustering Scheme

- Collection of objects (e.g., trajectories): $C = \{c_1, c_2, ..., c_M\}$
- Distance matrix: $D(i, j) = \delta(c_i, c_j)$

- Merge to create new cluster: $c_{ij} = \bigcup_{u \in \arg\min_{i,j,i \neq j} D_{i,j}} c_u$

- Birth index, $b_{ij} = D_{i,j}$, is distance threshold at which class $c_{ij}$ starts to exist
- Death index, $d_i = d_j = D_{i,j}$, is distance index when $c_i (c_j)$ ceases to exist

# Output of Hierarchical Clustering

- Tree structure, $\mathcal{T}\langle \mathcal{C}, \rho \rangle$

  $\mathcal{C}$ : collection of all original/hierarchical classes

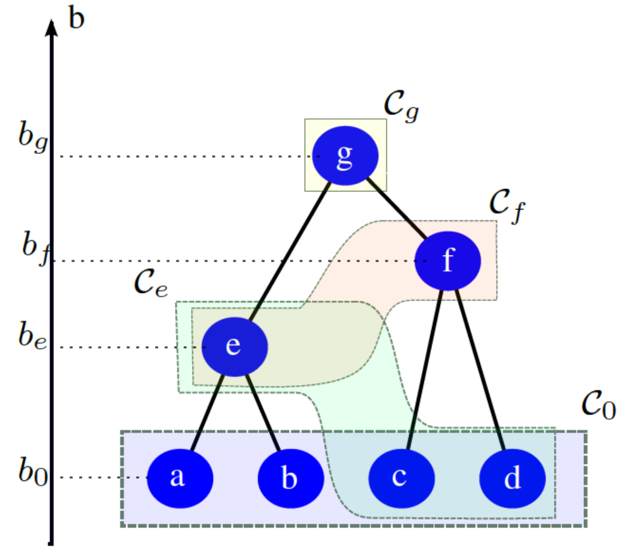
  $\rho : \mathcal{C} \mapsto \mathcal{C}$ : maps class to its 'parent'

- If $\rho(c_i) = c_j$

  then $b_j = d_i$ and $c_i \subset c_j$

- Tree node $c_i \in \mathcal{C}$ is alive when $b_i \leq b < d_i$

  denote the level by $\mathcal{C}_b$

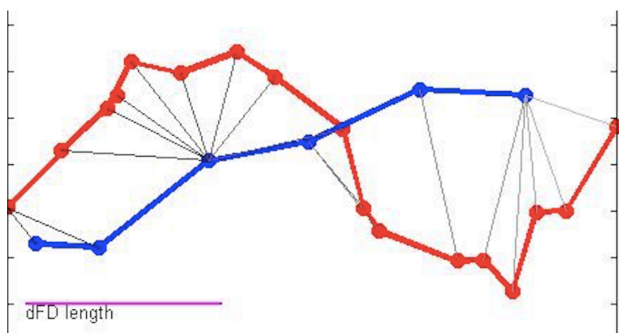


# Fréchet Distance

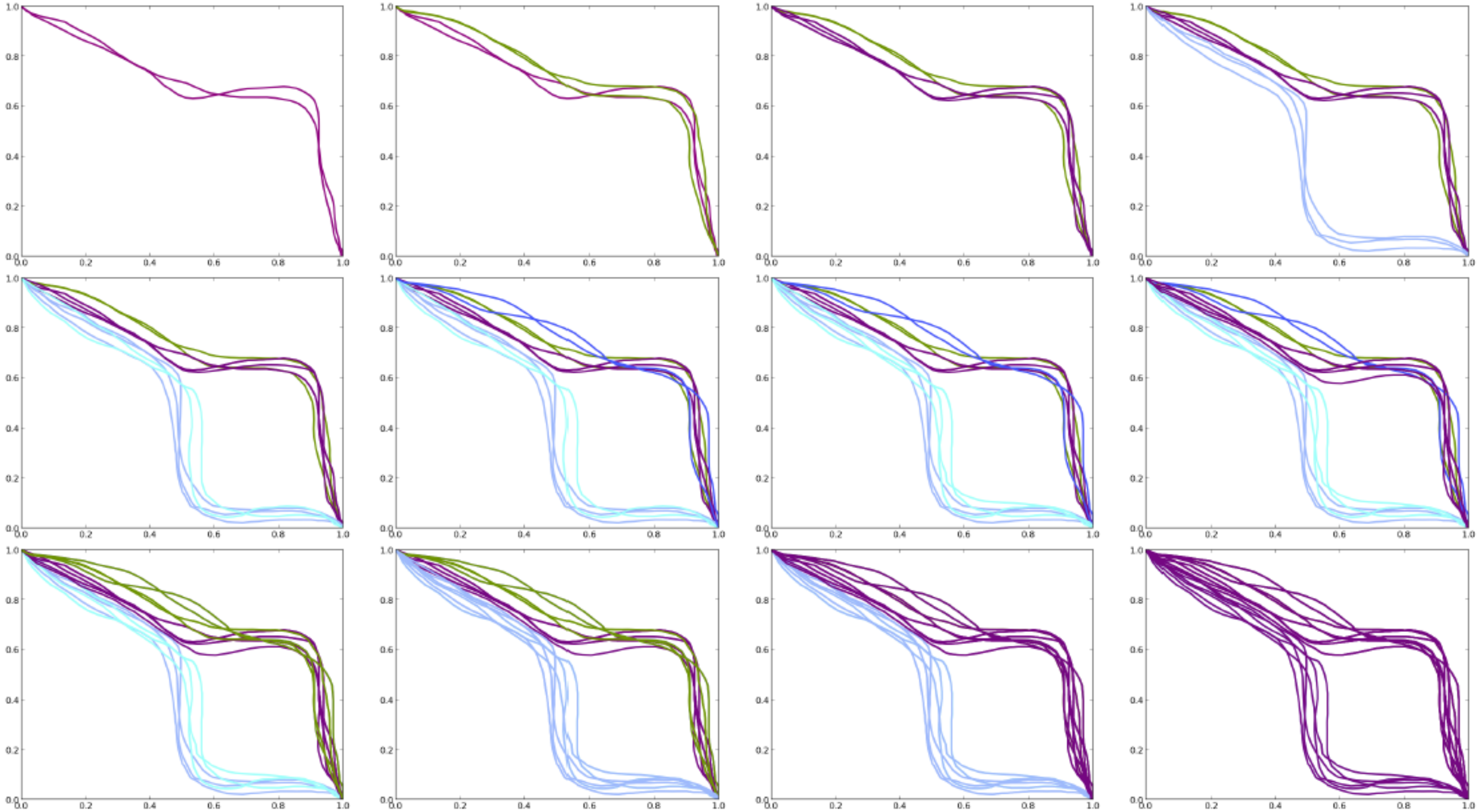
- Distance between two curves (or surfaces)
- The discrete Fréchet distance is defined as

$$\delta_F(\tau_1, \tau_2) = \inf_{\alpha,\beta} \max_j \delta_E(\tau_1(\alpha(j)), \tau_2(\beta(j)))$$

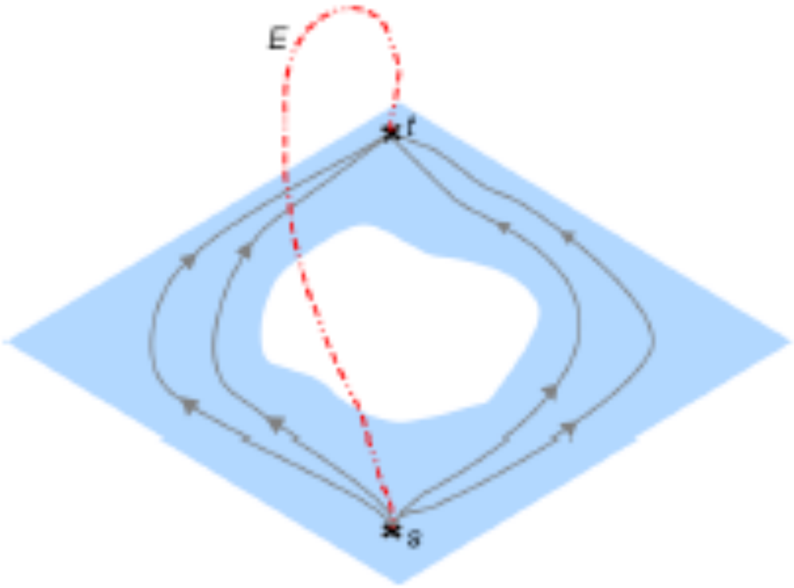
$\alpha, \beta$ are re-parametrisations that align trajectories to each other point-wise



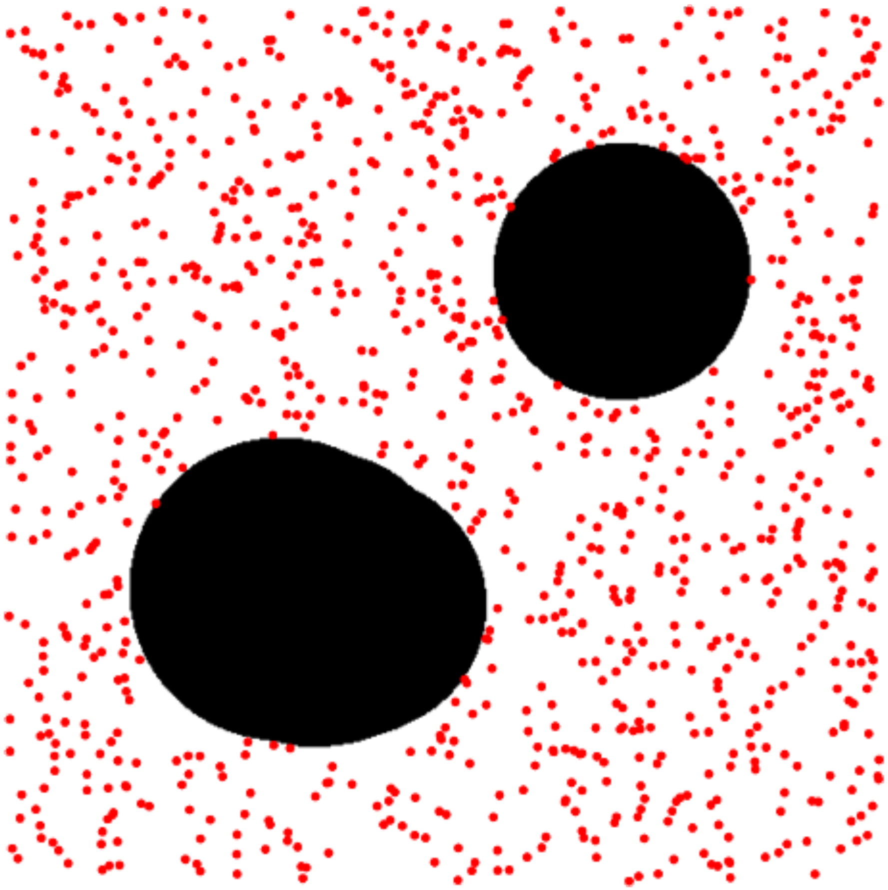
# Illustrative Example: Hierarchical Clustering with Fréchet Distance



# Clustering by Topology



# Recap: Constructing Filtrations from Samples



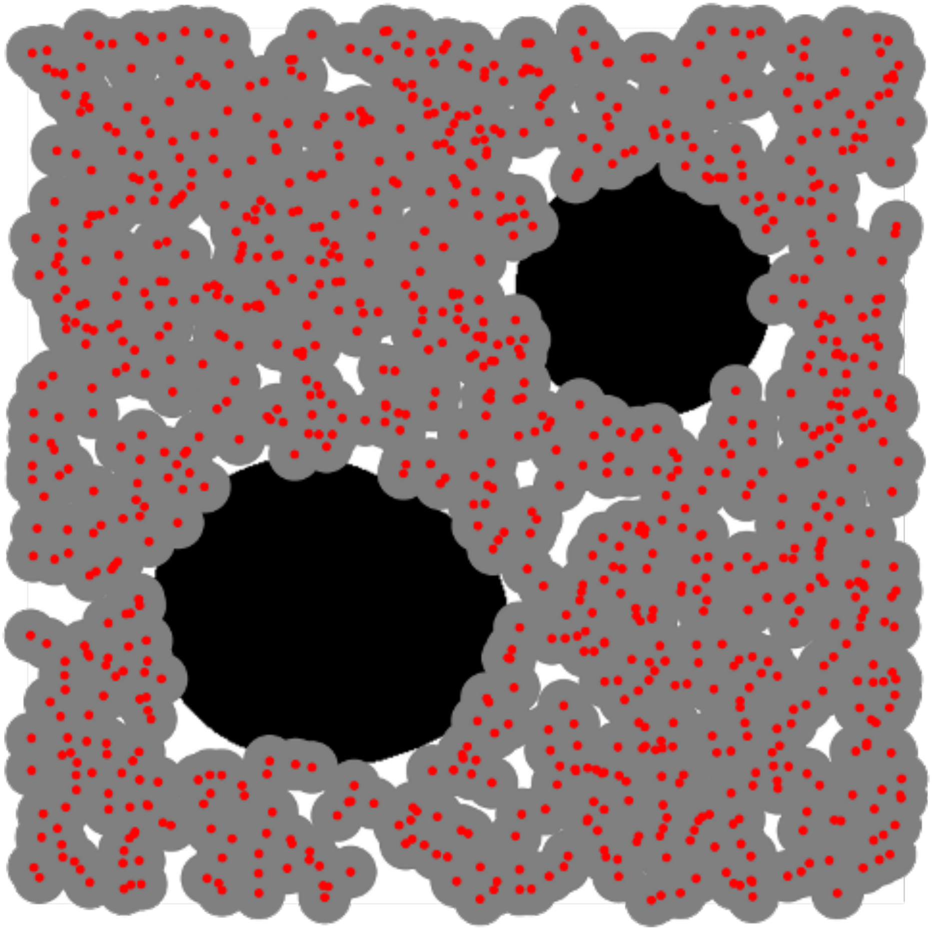
Observed trajectories yield samples in the C-Space,

$$X \subset \mathcal{C}_f \subset \mathbb{R}^d$$

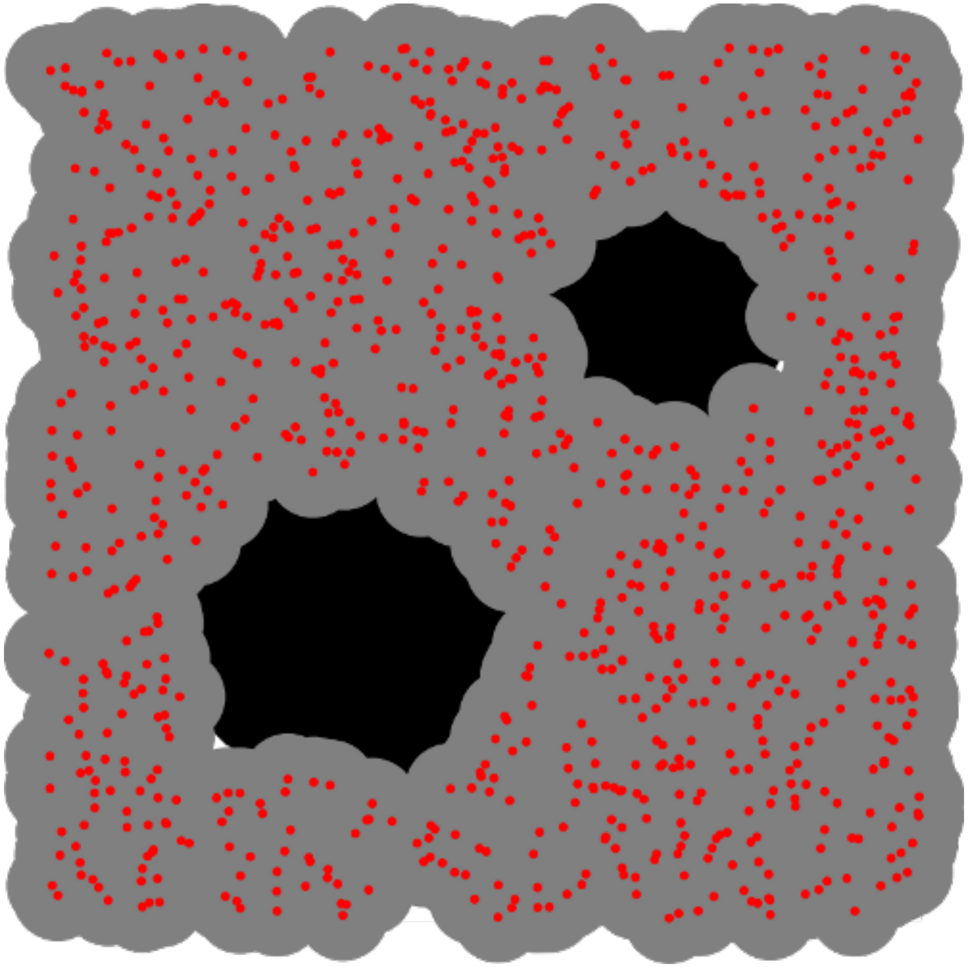
Then, consider the space,

$$X_r = \bigcup_{x \in X} \mathbb{B}_r(x)$$

# Unions of Balls



# Unions of Balls



# Delaunay – Čech Complex

$$D(X) = \{\sigma \subseteq X : \cap_{x\in\sigma} V_x \neq \emptyset\}$$

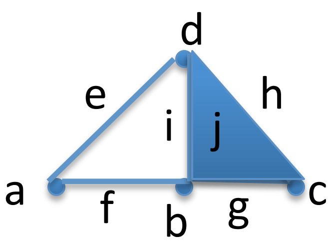
$$DC_r(X) = \{\sigma \in D(X) : \cap_{x\in\sigma} \mathbb{B}_r(x) \neq \emptyset\}$$

$$DC_r(X) \simeq X_r$$

If we can transform a simplicial complex K into another complex $K_0$ by a sequence of elementary collapses, then the two complexes have the same homotopy type.
A discrete Morse function can encode such a simplicial collapse, which is used to establish the above result.

U. Bauer, H. Edelsbrunner. **The Morse theory of Čech and Delaunay filtrations**.
In Proc. Symp. Comp. Geometry, SOCG'14.

# Homology Computation by Matrix Reduction



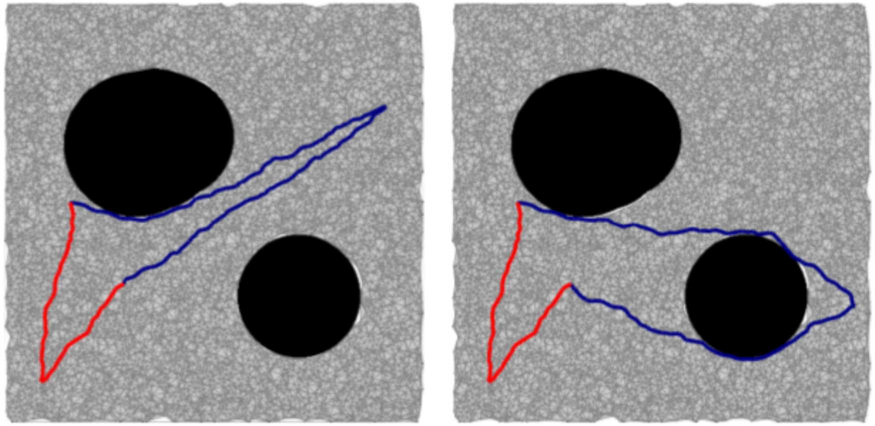

$g+h+i = [0\ 0\ 0\ 0]^T$

$h+i = [0\ 1\ 1\ 0]^T$

| | a | b | c | d | e | f | g | h | i | j |
|---|---|---|---|---|---|---|---|---|---|---|
| a | | | | | 1 | 1 | 0 | 0 | 0 | |
| b | | | | | 0 | 1 | 1 | 0 | 1 | |
| c | | | | | 0 | 0 | 1 | 1 | 0 | |
| d | | | | | 1 | 0 | 0 | 1 | 1 | |
| e | | | | | | | | | | 0 |
| f | | | | | | | | | | 0 |
| g | | | | | | | | | | 1 |
| h | | | | | | | | | | 1 |
| i | | | | | | | | | | 1 |
| j | | | | | | | | | | |

[H. Edelsbrunner, J. Harer, *Contemp. Math.* **453**: 257-282, 2008.]

# Trajectory Classification in Simplicial Complex



$$\alpha_0, \ldots, \alpha_n \in C_1(DC_R(X))$$

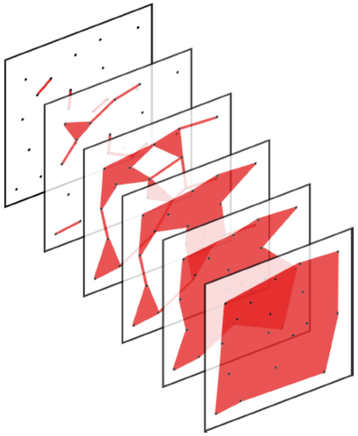
$$c_{\alpha_0}(\alpha_j) = \alpha_0 + \alpha_j \in Z_1(DC_R(X))$$

$$[c_{\alpha_0}(\alpha_0)], [c_{\alpha_0}(\alpha_1)] \ldots, [c_{\alpha_0}(\alpha_n)] \in H_1(DC_R(X))$$

$[c_{\alpha_0}(\alpha_i)] \neq [c_{\alpha_0}(\alpha_j)] \implies \alpha_i,\ \alpha_j$ are not homotopy equivalent in $X_R$

# **Persistent** Homology Groups

$$DC_{r_1}(X) \subseteq DC_{r_2}(X) \subseteq \ldots \subseteq DC_{r_n}(X)$$



For a **filtration of simplicial complexes**

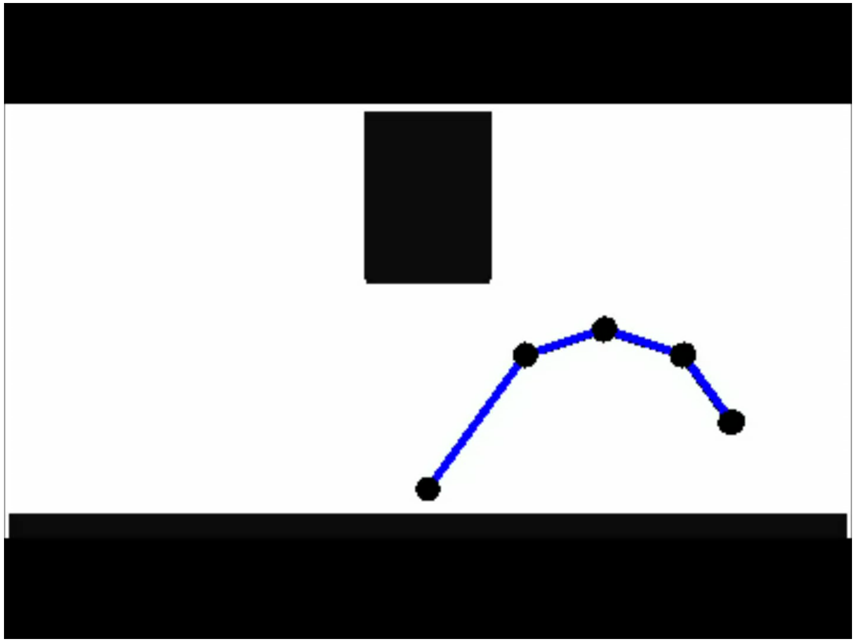
$$\mathcal{K}_{r_1} \subseteq \mathcal{K}_{r_2} \subseteq \ldots \subseteq \mathcal{K}_{r_n} \quad ,$$

define the **p-th persistent homology** groups:

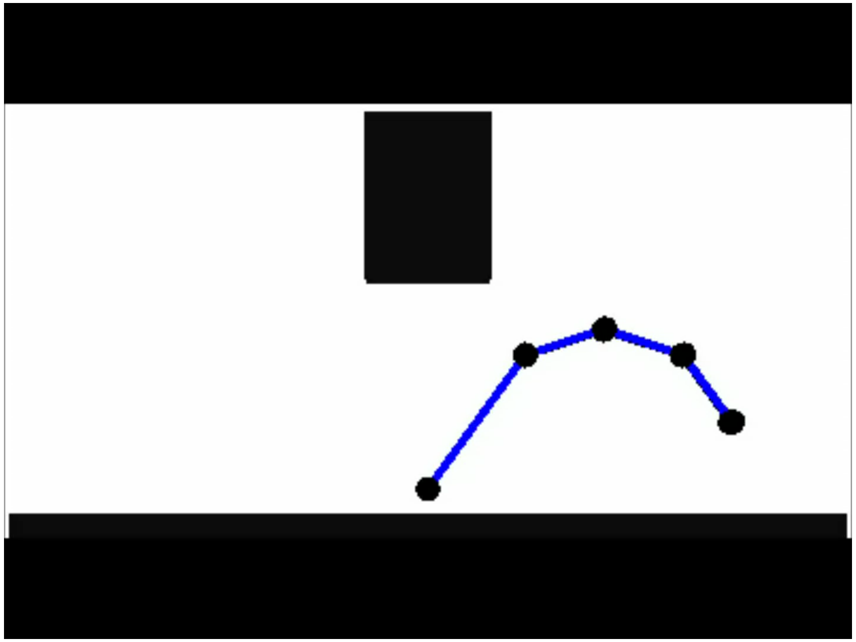
$$H_p^{i,j} = Z_p(\mathcal{K}_{r_i})/(B_p(\mathcal{K}_{r_j}) \cap Z_p(\mathcal{K}_{r_i})) \;\; i \leqslant j$$

# Two Trajectory Classes for 4-link Arm

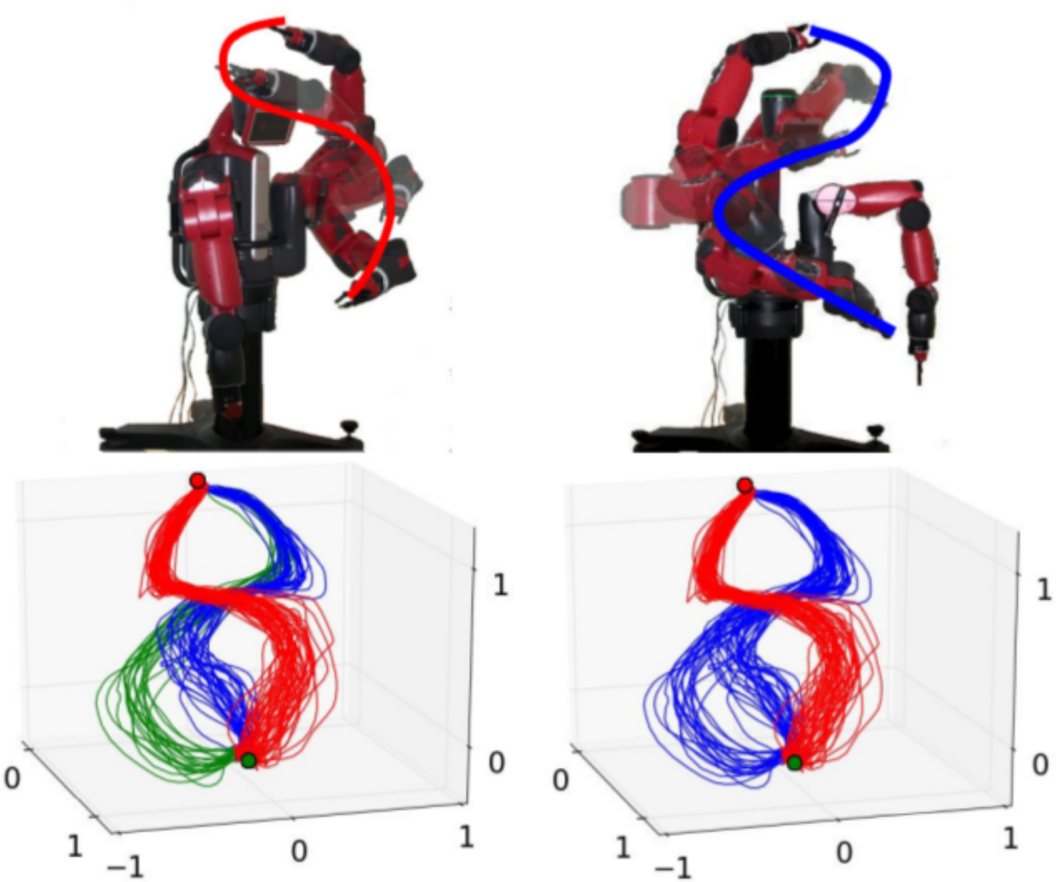
**Class “S”**



**Class “Z”**



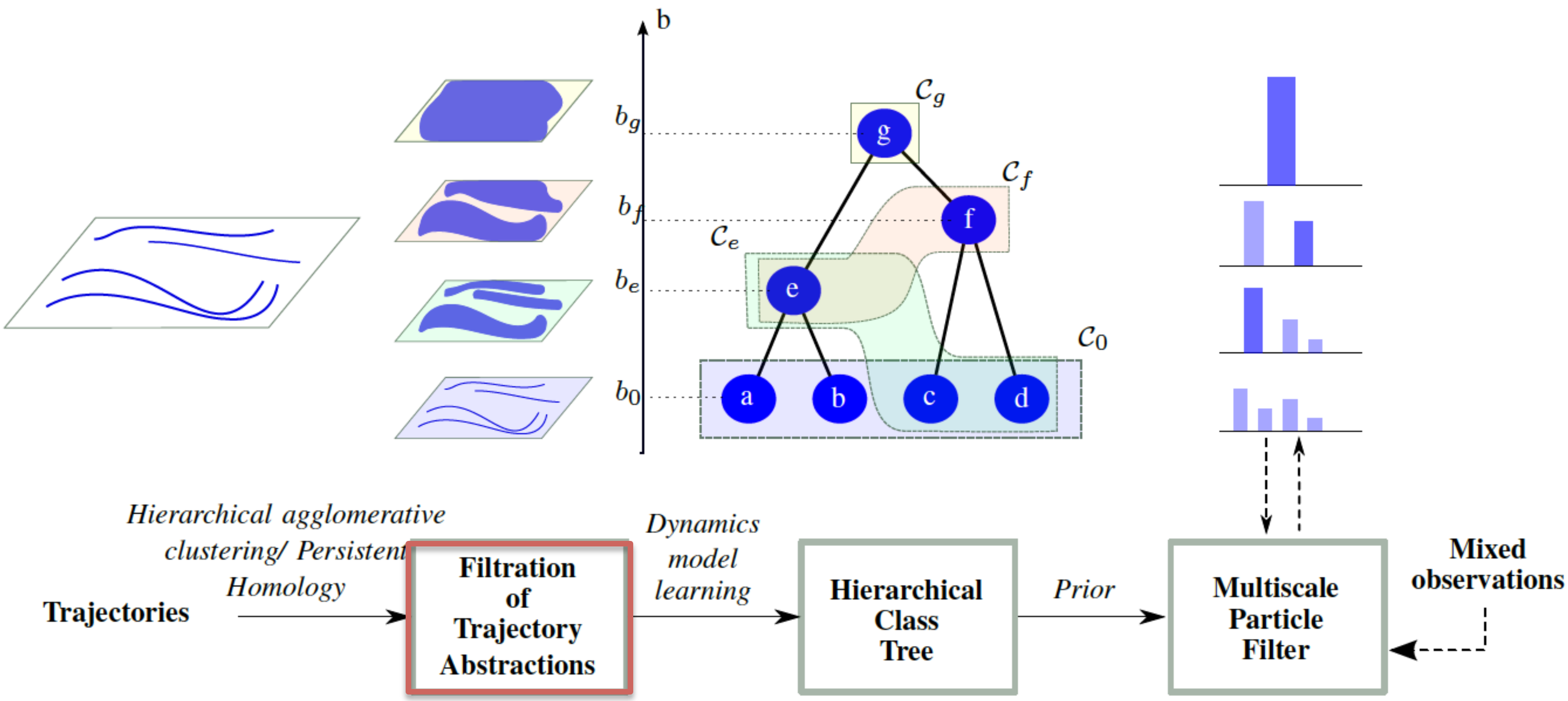
# Clustering End Effector Trajectories: 3-dim, 97 trajectories




There exists a filtration interval with 3 classes which later merge into two classes

# Two Discovered Equivalence Classes

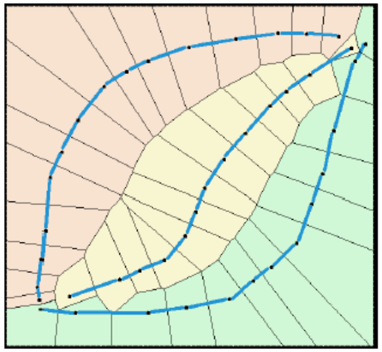
# Recap: Approach



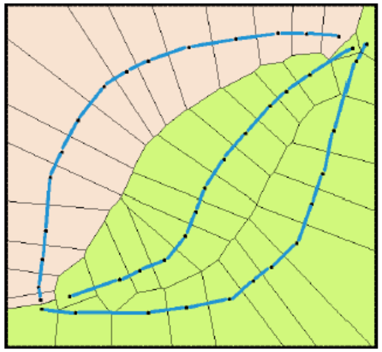
# Hierarchical Beliefs - Intuition

- Probability of nodes in the tree ⇔ probabilities assigned to regions in a metric space
- Intuition: Consider Voronoi tessellation of 2-dim space by points on discrete curves

$$\mathbf{P}(c_j) = \sum_{c_i=\rho^{-1}(c_j)} \mathbf{P}(c_i)$$



**(a)** Three separate classes



**(b)** Two classes merge

# Bank of Particle Filters (MPF)

1. Sample $n$ particles with equal weights from a prior,

$$\Delta(\mathcal{C}_0 \times \Re^d) \qquad\qquad \mathcal{C}_0 = \{c \in \mathcal{C} | b_c = b_0\}$$

2. Build tree probabilities*
3. Sample new particles for parent nodes: $\bar{n} = \sum_{c_i = \rho^{-1}(\bar{c})} n_i$

In a loop:

a. For each particle at each level, compute new position:

$$z^t \sim \mathbf{P}(z | z^{t-1}, c)$$

b. For particles at chosen level, update with observations*

c. Propagate weights along tree & resample (at lowest level)*

# Motion Models

- A motion model is learnt for every class (tree node)

$$\theta_i = \mathbf{P}(z'|z, c_i),\ \ z, z' \in \Re^d$$

- The collection of trajectories define:
  - Either a global model, e.g., using GMM/GMR
  - Or, a localised model - the average velocity of the trajectory points in a ball around the studied point z

$$\dot{z} = \textstyle\sum_{z_i \in \text{ball around z}} \ 1/\delta_E(z, z_i) \ \ \dot{z}_i$$

# Incorporating Evidence, Base Case

- At level 0 (i.e., at the level of the finest scale), compute Euclidean distance between observed position and class prediction $\delta_E(z, \xi^t)$

- Update weights, inversely proportional to distance

$$p(z^t, c) : \; w \propto -\log(\delta_E(z, \xi^t))$$

# Incorporating Evidence, with Coarse Observations

First find all classes that are alive at the level of observation,

$$\hat{\mathcal{C}} = \{c_i \in \mathcal{C} |\ b_i \leq \frac{b_\xi + d_\xi}{2} < d_i\}$$

For each of these classes:

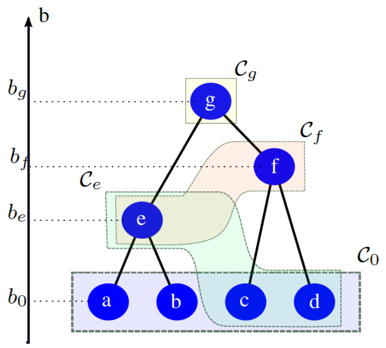
Compute distance based on birth index of first shared parent

$$\delta_\mathcal{T}(c, \xi^t) = \hat{b}$$

For every particle in class $c$, update as

$$p(z^t, c) :\ \ w \propto -\log(\delta_\mathcal{T}(c, \xi^t))$$

Rebuild tree probabilities and normalise



# Rebuilding Probabilities in the Tree

1. Update based on computed weights, $w$, & node probability,

$$\mathbf{P}^t(c_i) = \frac{\sum_{c(p)=c_i} w^t}{\sum_{c(p)\in\hat{\mathcal{C}}} w^t}$$

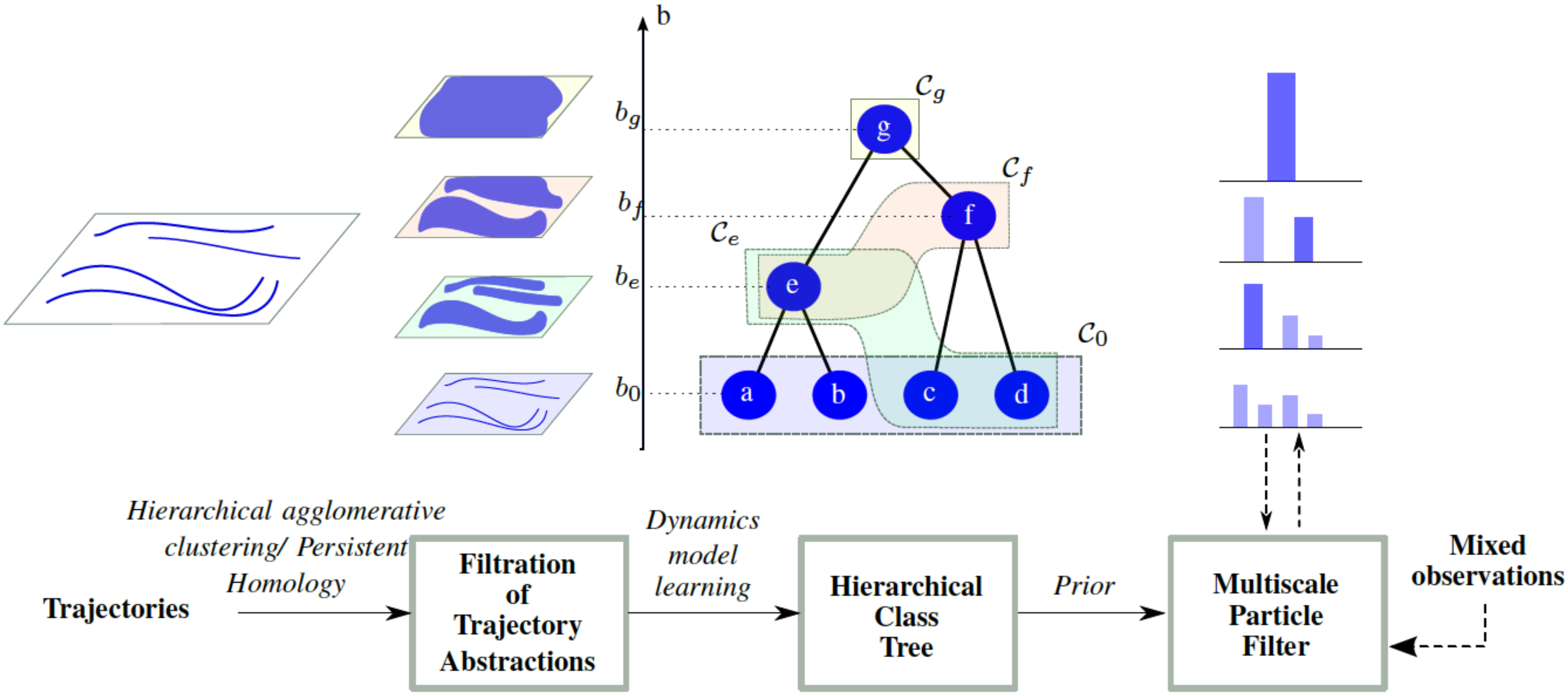
2. Downward pass: Recursively update childrens' probabilities

$$\mathbf{P}^t(\underline{c}) = \mathbf{P}^{t-1}(\underline{c})\frac{\mathbf{P}^t(\rho(\underline{c}))}{\mathbf{P}^{t-1}(\rho(\underline{c}))}$$

3. Upward pass: Recursively update parents' probabilities

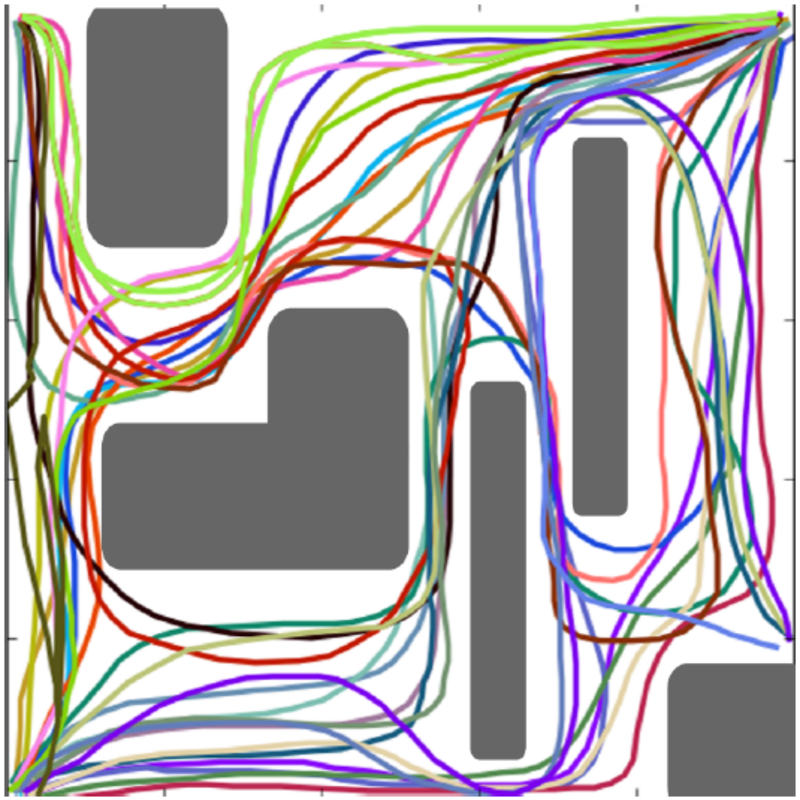
$$\mathbf{P}^t(\overline{c}) = \sum_{c=\rho^{-1}(\overline{c})} \mathbf{P}^t(c)$$

# Recap: Approach

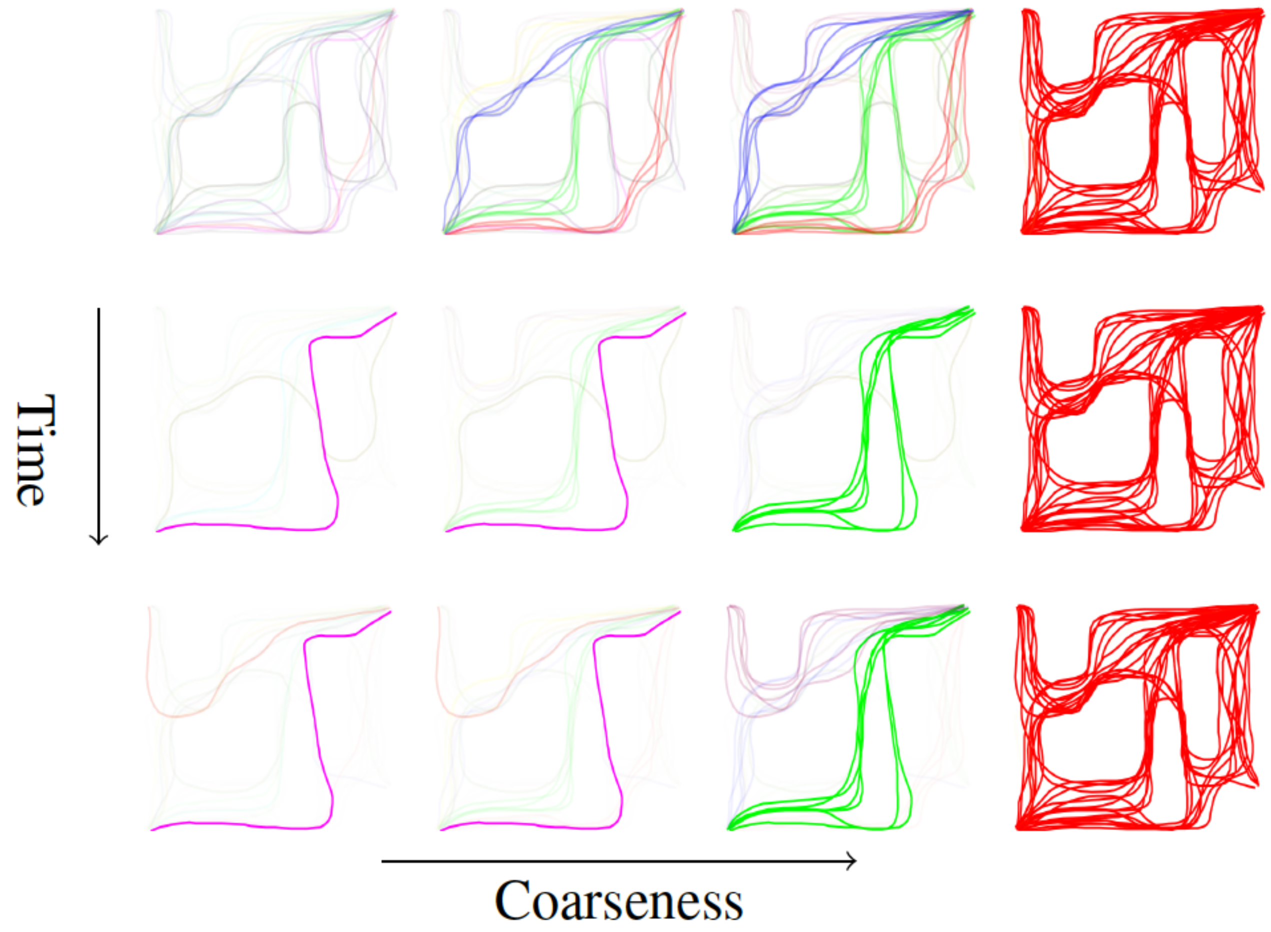


# Experiment 1: Qualitative Behaviour

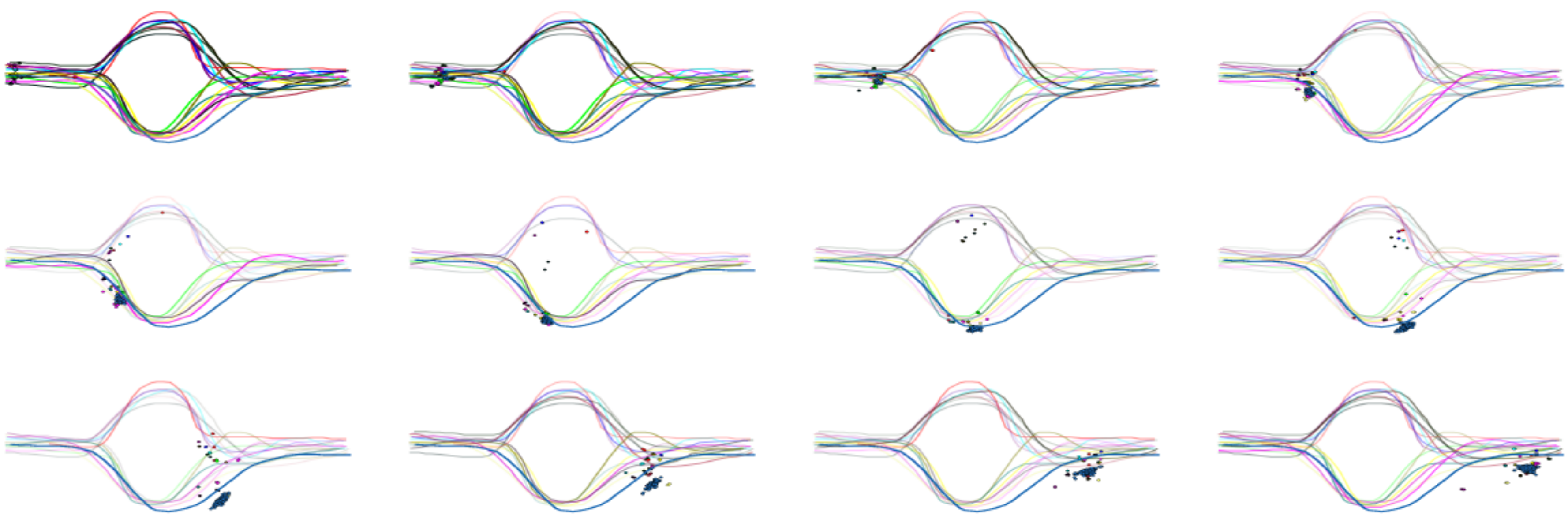
- We characterise the temporal behaviour using a synthetic dataset
- The data mimics typical pattern of movement of pedestrians in built environments
- We update all probabilities in the tree



# Results: Beliefs Over Time and Level



# Time Evolution of Beliefs



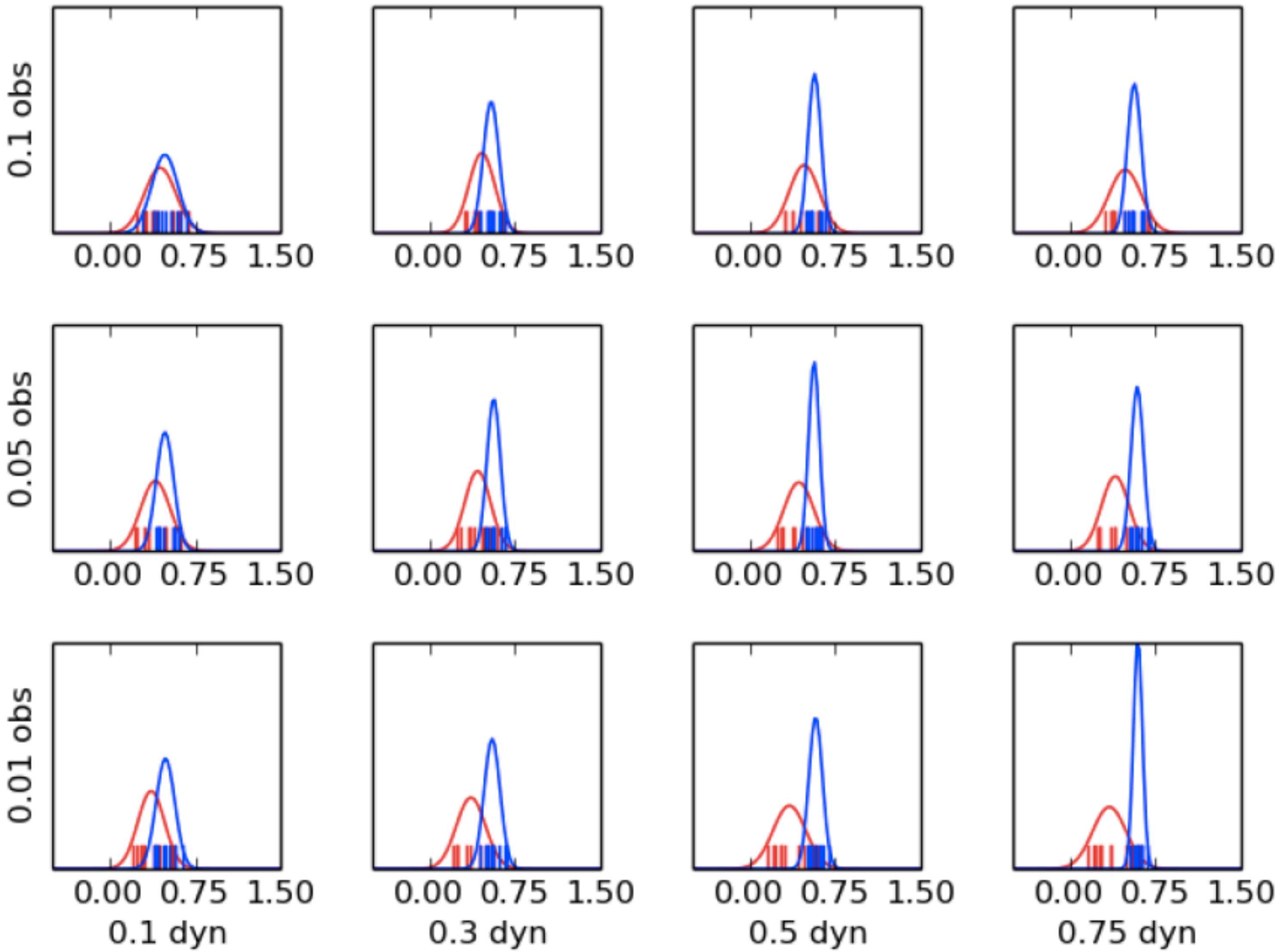
# Experiment 2: Understanding Performance

MPF vs. Baseline PF

Average normalised distance from true trajectory

- Metric: distance between maximum a posteriori class of the filter and the true trajectory, as captured by the tree distance, averaged over time
- Coarse observations are provided uniformly at random (50% of the time there will also be a coarse observation)

# MPF vs. Baseline



# Experiment 2: Understanding Performance

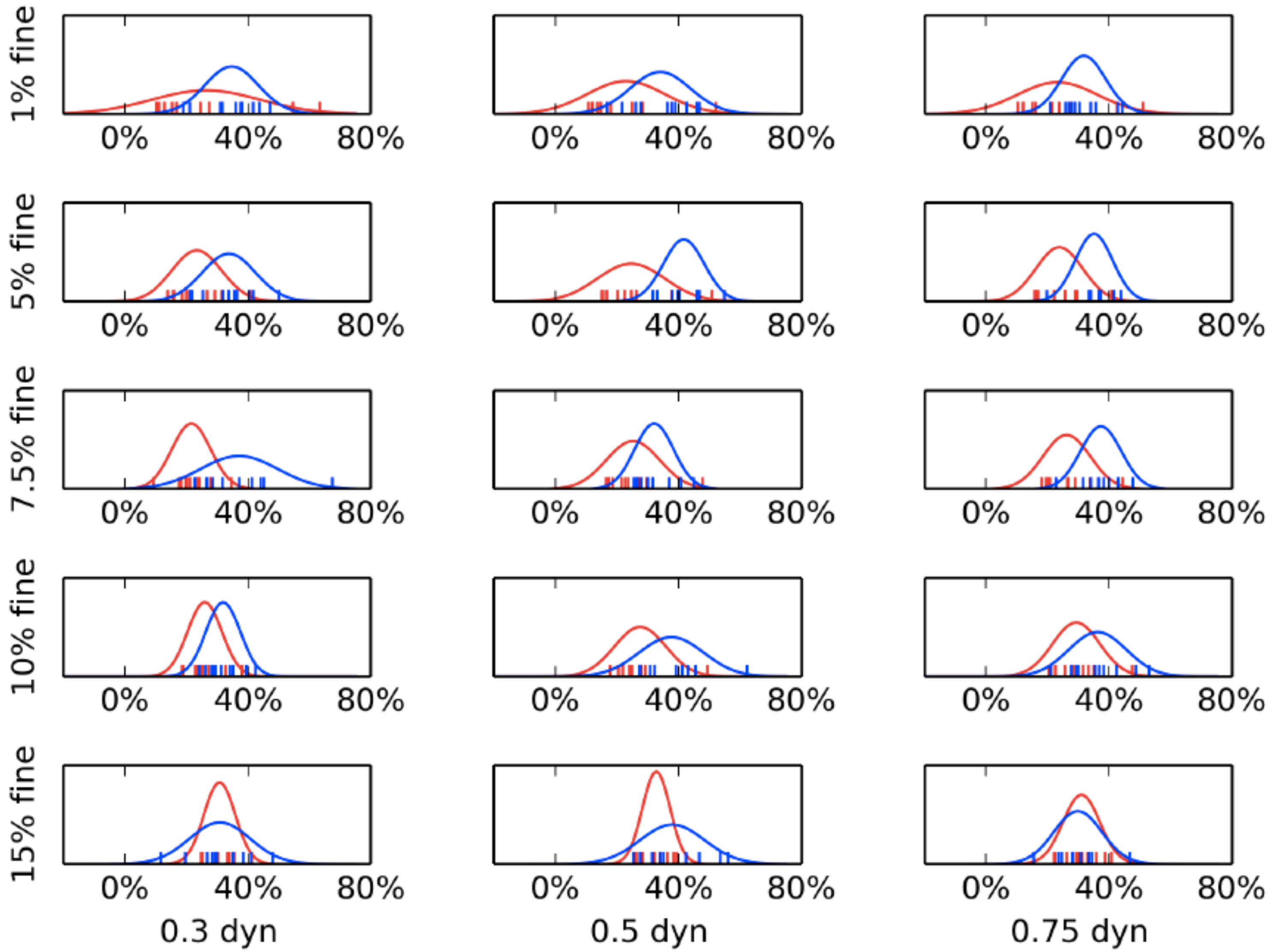
MPF vs. Baseline PF

Time to convergence

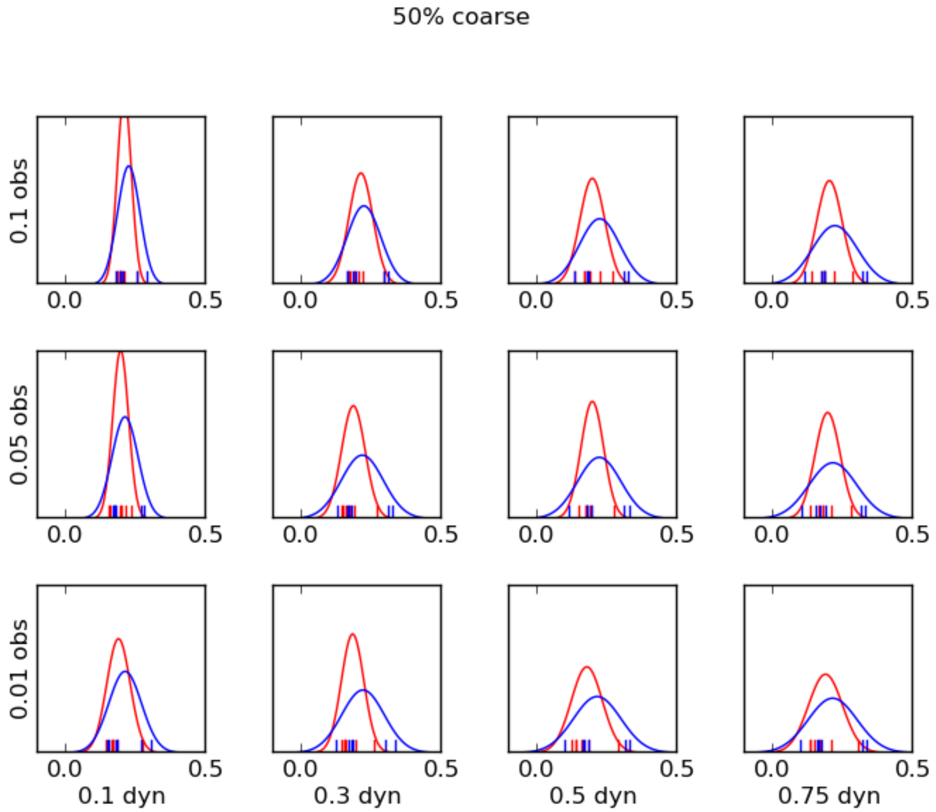
- Metric: normalised time taken (%) for estimate to be within $\varepsilon$, by similarity in tree distance, of true trajectory.

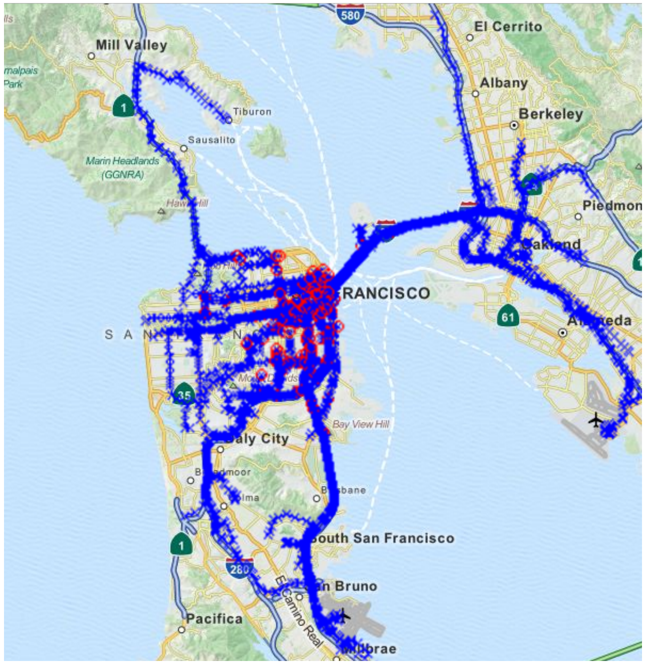
- Both versions receive fine observations for a small burn-in period followed by only coarse observations (which baseline PF is unable to use)

# MPF vs. Baseline

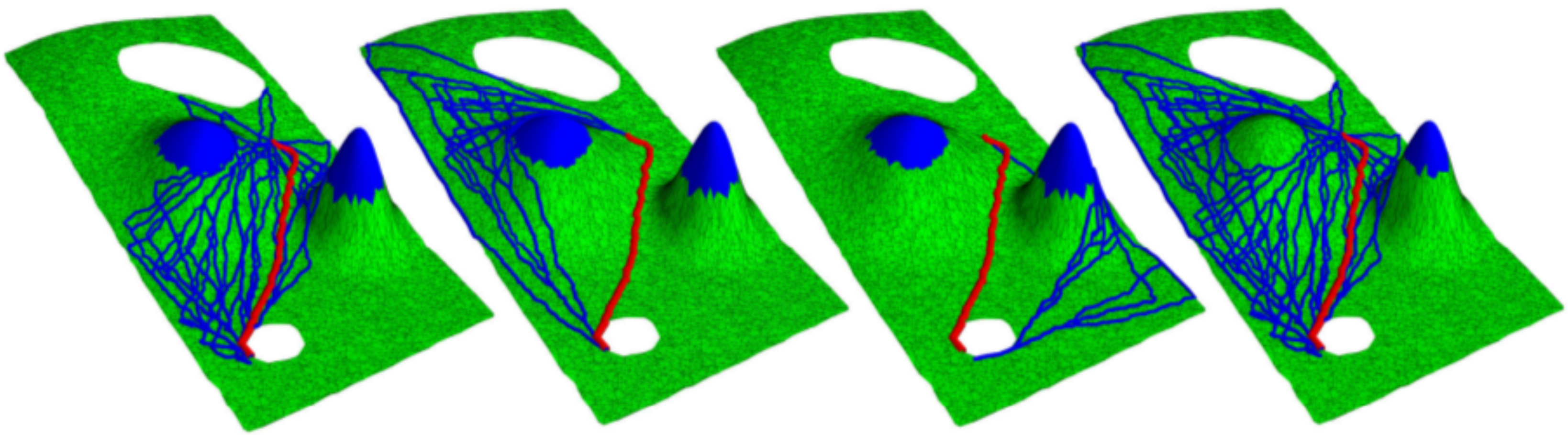


# Ongoing Experiment with Uber Dataset



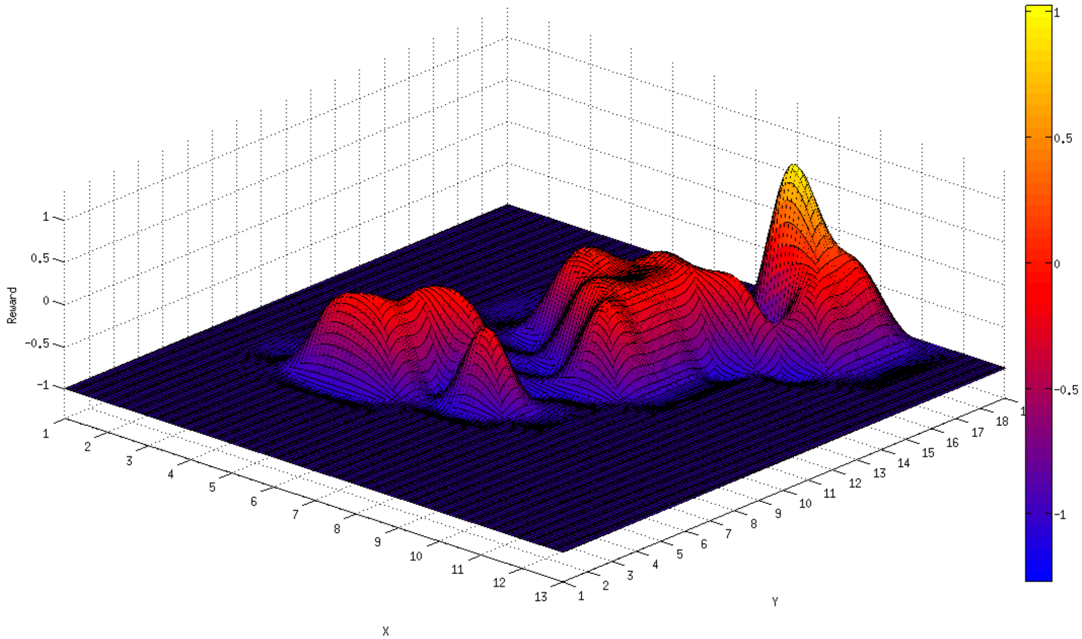
# Recap: Filtrations Arising from Sublevel Sets



With general models such as Gaussian Processes, does this approach offer leverage by extracting qualitative structure to make learning more tractable?

# Example Surfaces of Interest: MDP Activity Models

[F. Previtali et al., ICRA 2015, IROS 16 subm]




Inverse RL methods can be used to estimate cost functions

Realised paths are actually determines by a value function, expensive computation:

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} \mathbf{P}^{a}_{ss'} \left[ \mathbf{R}^{a}_{ss'} + \gamma \; V^{\pi}(s') \right]$$

Can topological categorisation of reward functions yield faster computation of V?

# Conclusions

- Persistence as defined in TDA provides a multi-scale representation that is ripe for combinations with probabilities
- Some uses:
  - Var. noise sensors/ actuators
  - Coarse instructions

# Acknowledgements

This work was supported by the EU projects
TOMSY (IST-FP7-270436) and TOPOSYS (ICT-FP7-18493)

We have benefited from the use of equipment within the
EPSRC *Robotarium* Research Facility at the Edinburgh Centre for Robotics.