Unfreezing the Robot: Navigation in Dense Human Crowds

Pete Trautman, Andreas Krause, Jeremy Ma and
Richard M. Murray

Why is Autonomous Crowd Navigation Needed?

- Malls, hospitals, ...

Why is Autonomous Crowd Navigation Needed?

- Malls, hospitals, ...

Why is Autonomous Crowd Navigation Needed?

- Malls, hospitals, ...
- Cafeterias!

Pioneer 3-DX

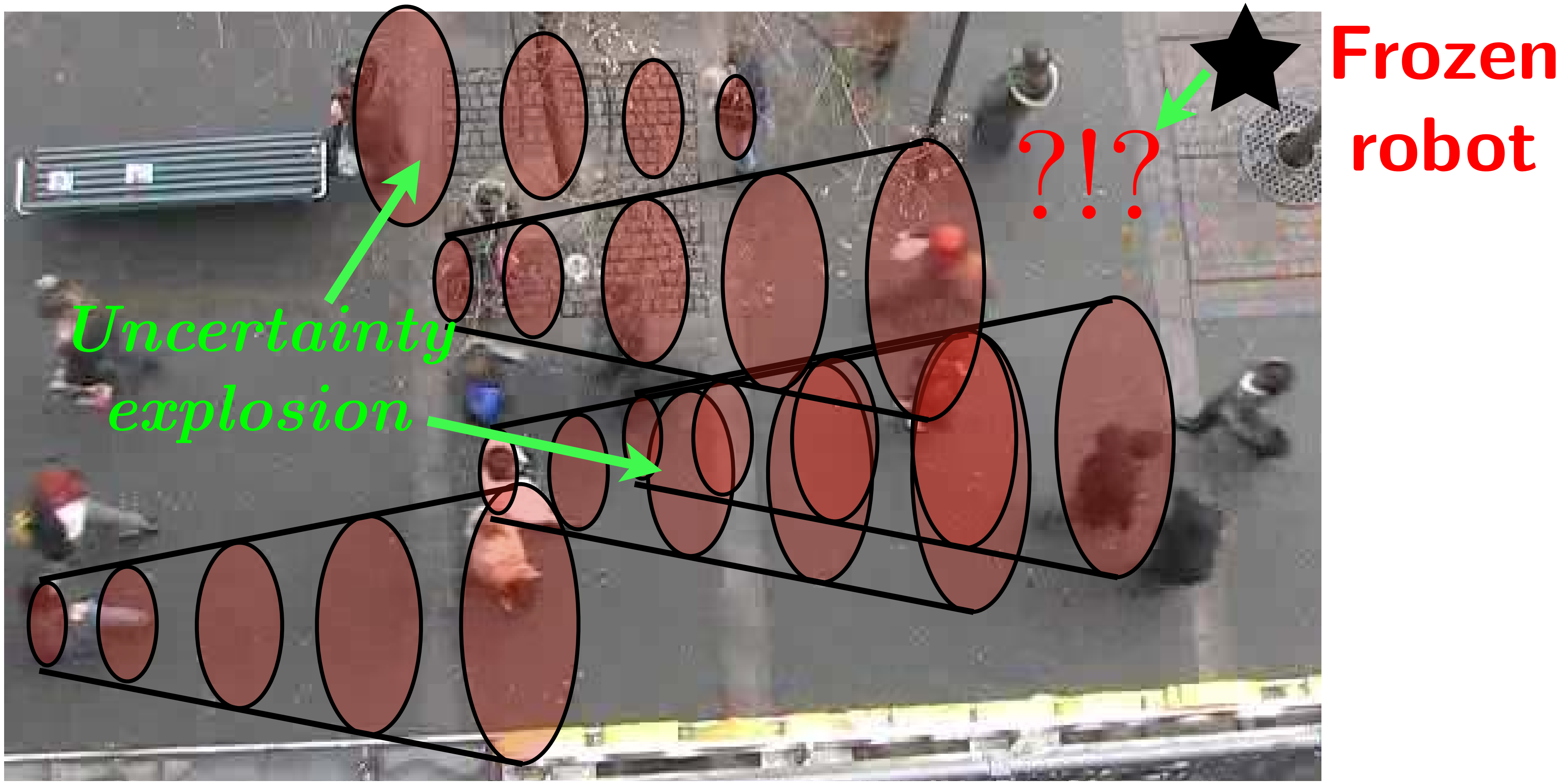
The Freezing Robot Problem

The Freezing Robot Problem

Independent agents $\implies$ uncertainty explosion

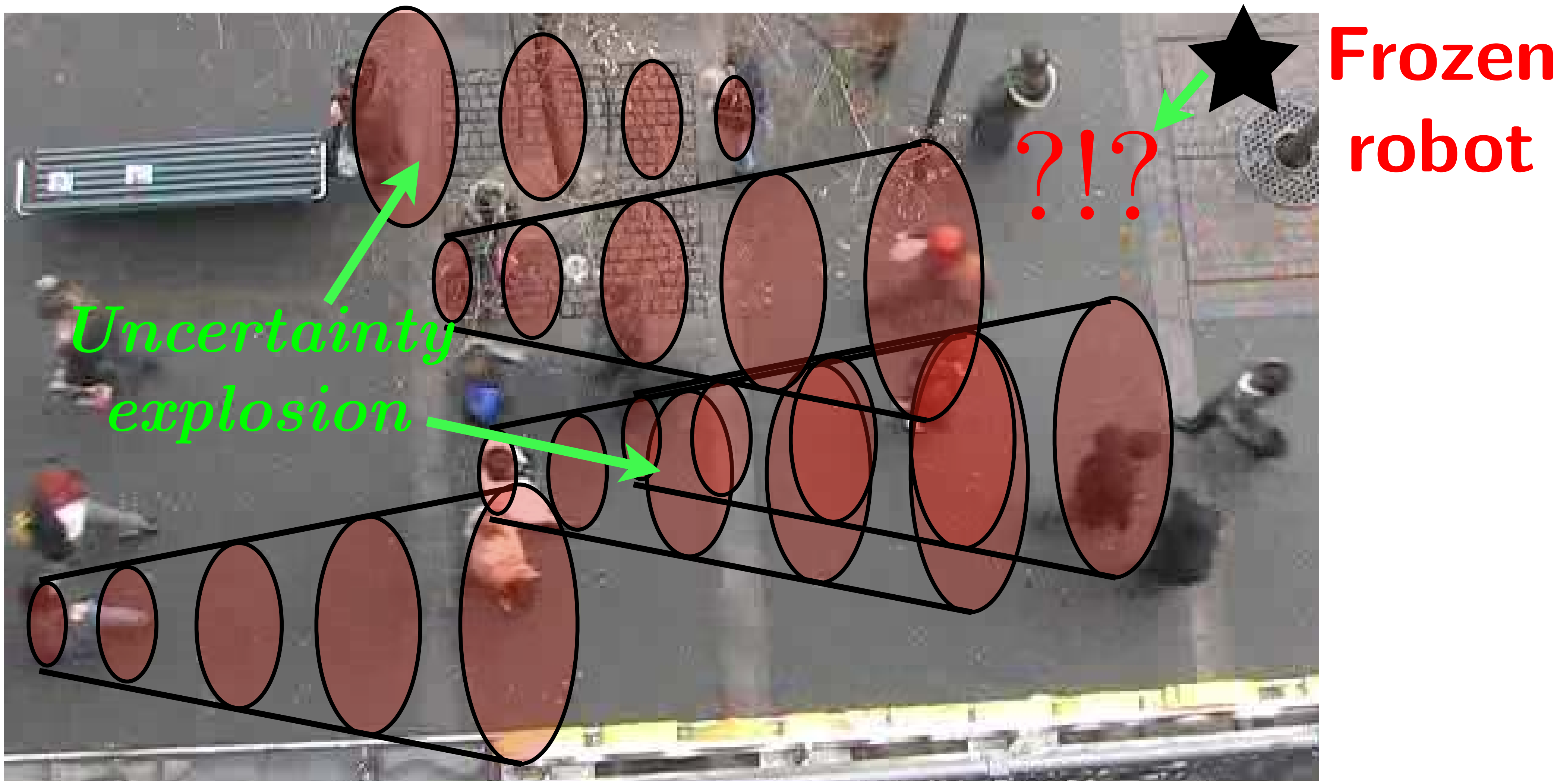
The Freezing Robot Problem

Independent agents $\Longrightarrow$ uncertainty explosion



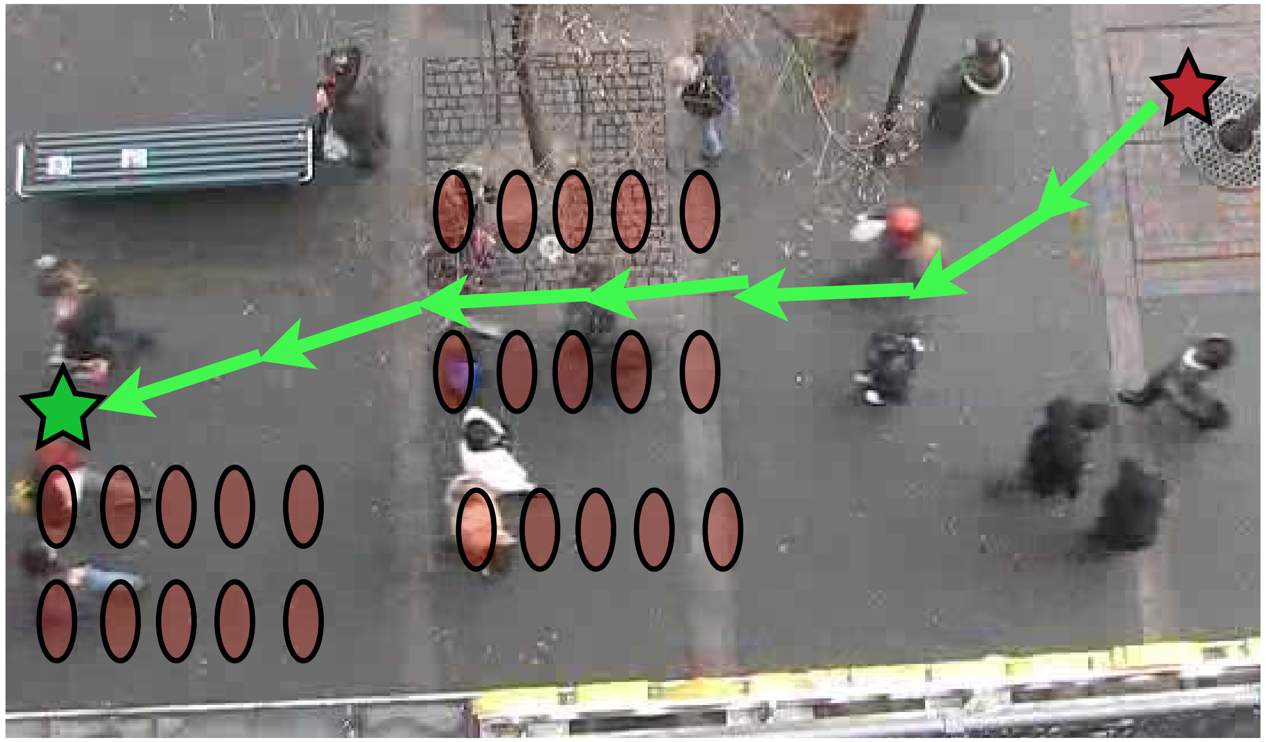
The Freezing Robot Problem

Independent agents $\implies$ uncertainty explosion



Uncertainty explosion $\implies$ **freezing robot problem**

Approaches to Solving the Freezing Robot Problem

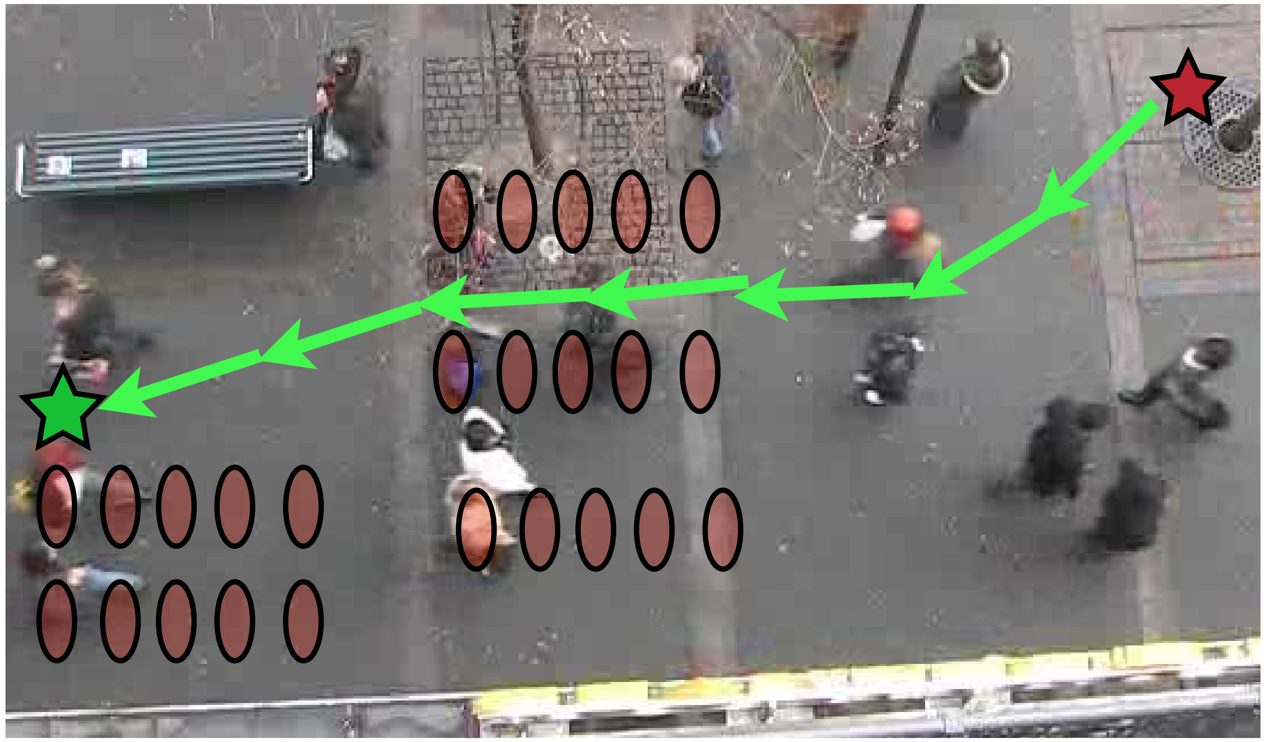


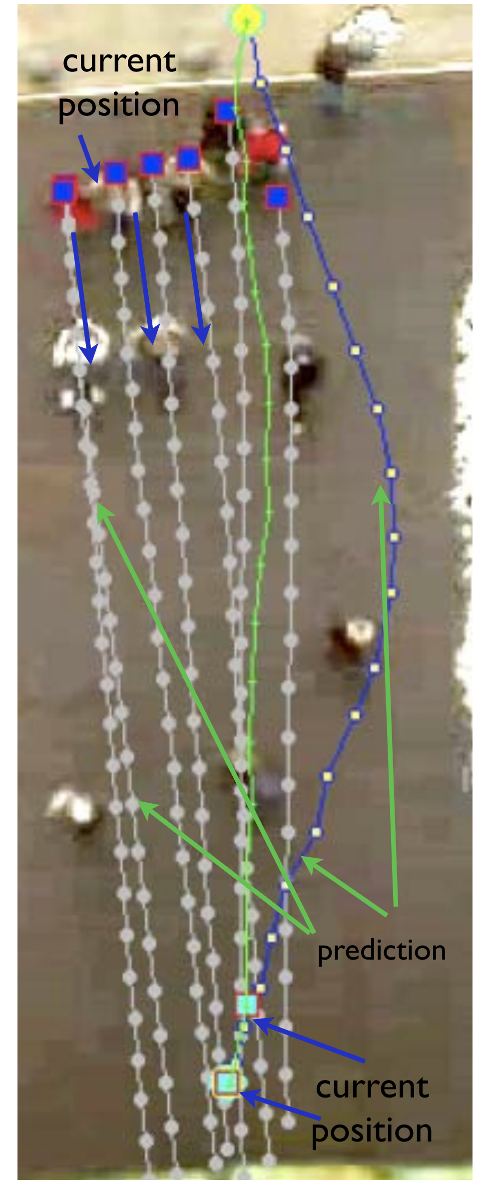
- State of the art methods assume culprit of FRP is uncertainty explosion [6]

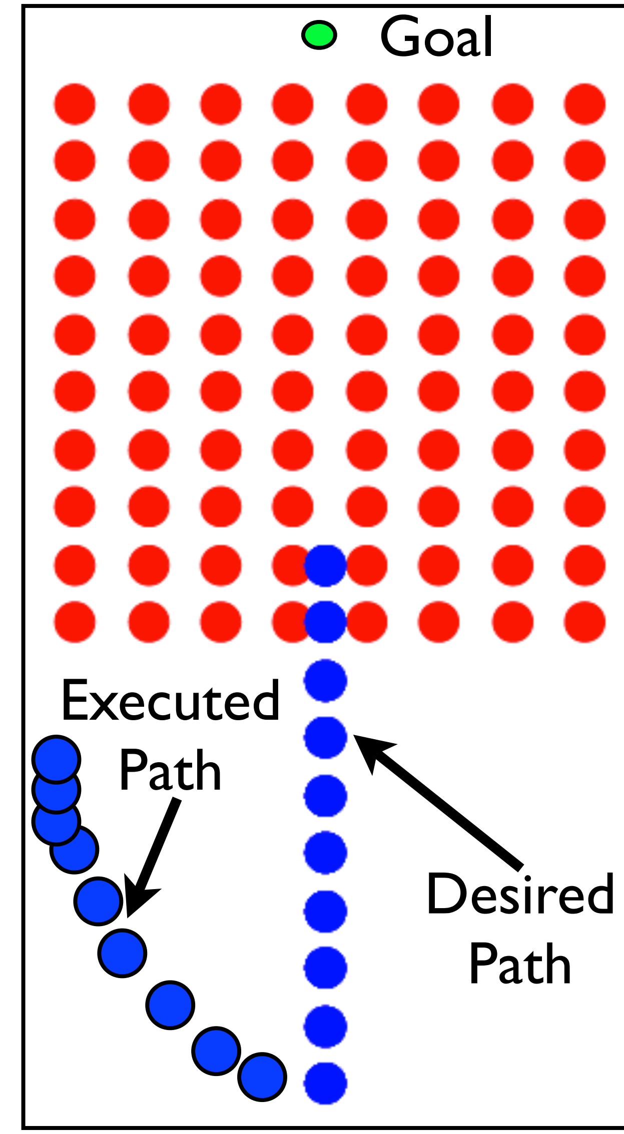
- Control covariance, keep cost low (call it "constant covariance" method)

- Precise agent dynamic modeling has same motivation [18,16,2,7,9]

Approaches to Solving the Freezing Robot Problem



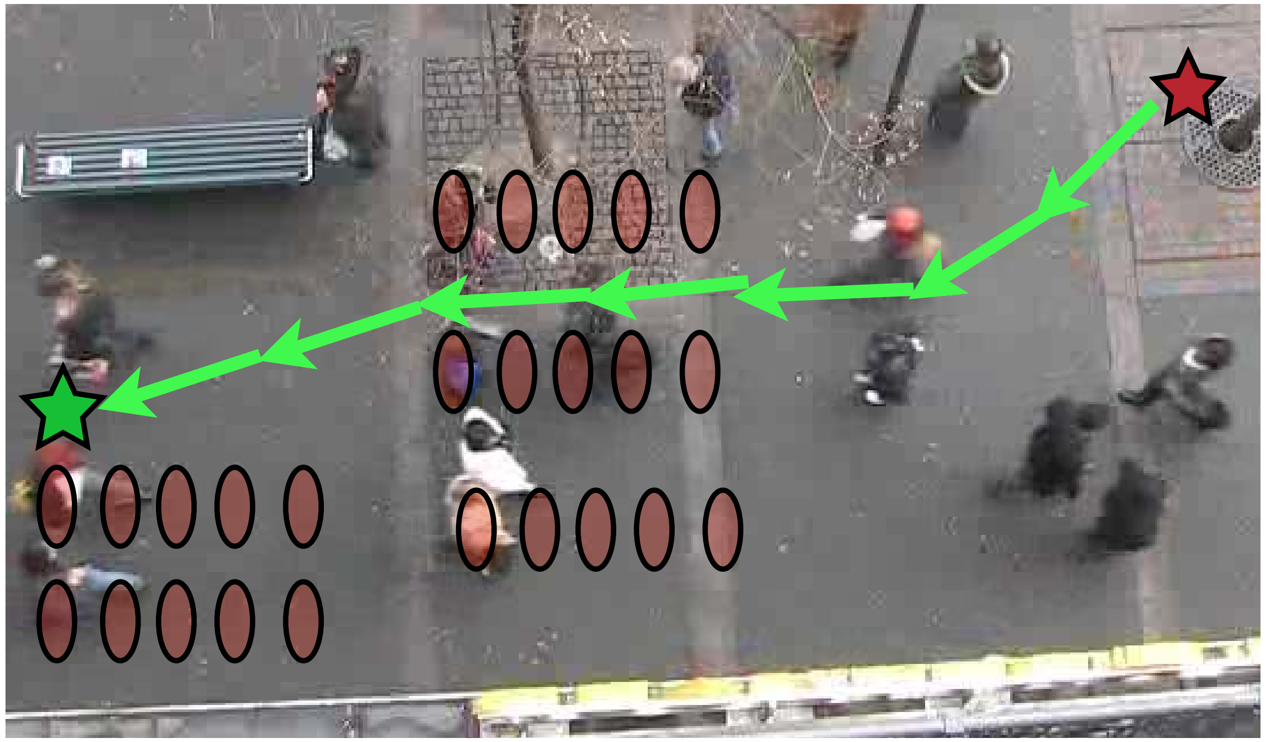




- State of the art methods assume culprit of FRP is uncertainty explosion [6]
- Control covariance, keep cost low (call it "constant covariance" method)
- Precise agent dynamic modeling has same motivation [18,16,2,7,9]

BUT: Severely crowded environments: all paths lower bounded by large cost

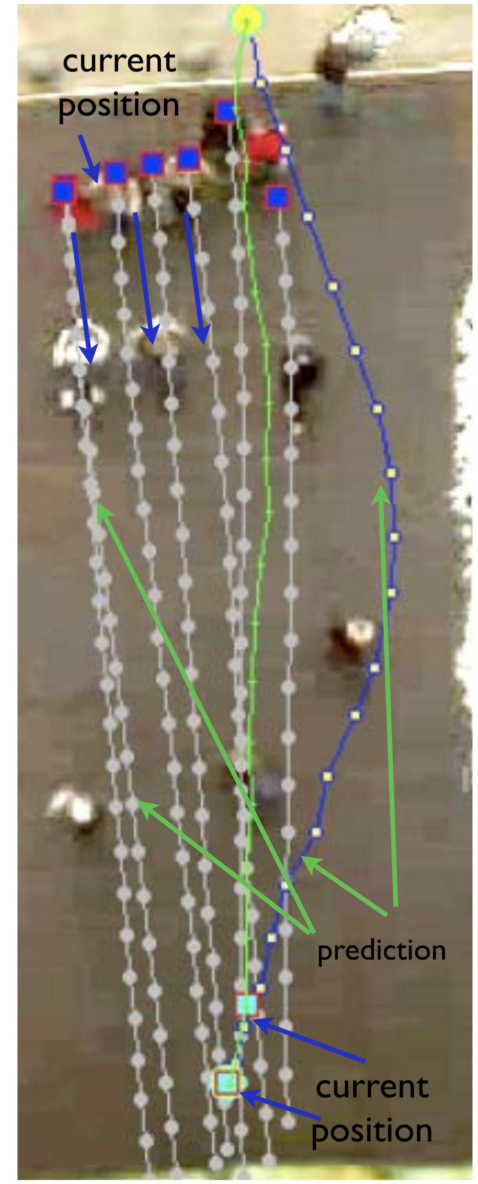
Approaches to Solving the Freezing Robot Problem

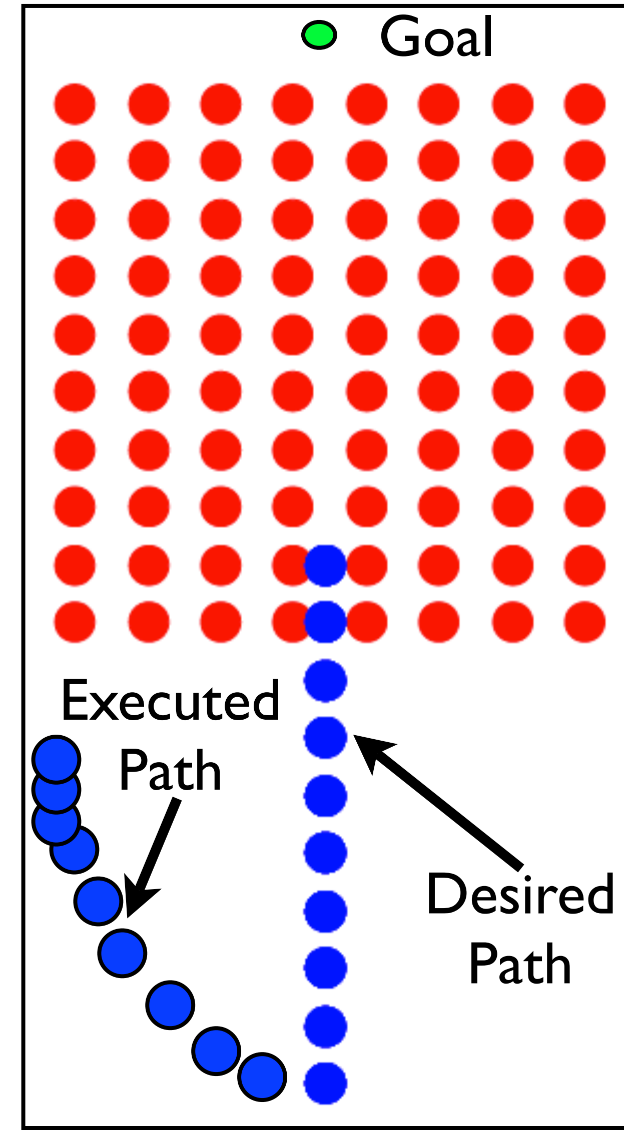


- State of the art methods assume culprit of FRP is uncertainty explosion [6]

- Control covariance, keep cost low (call it "constant covariance" method)

- Precise agent dynamic modeling has same motivation [18,16,2,7,9]



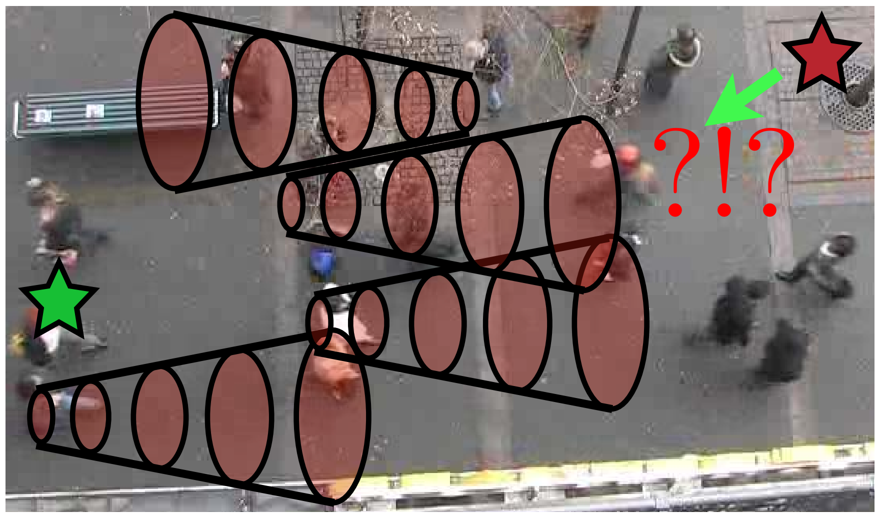


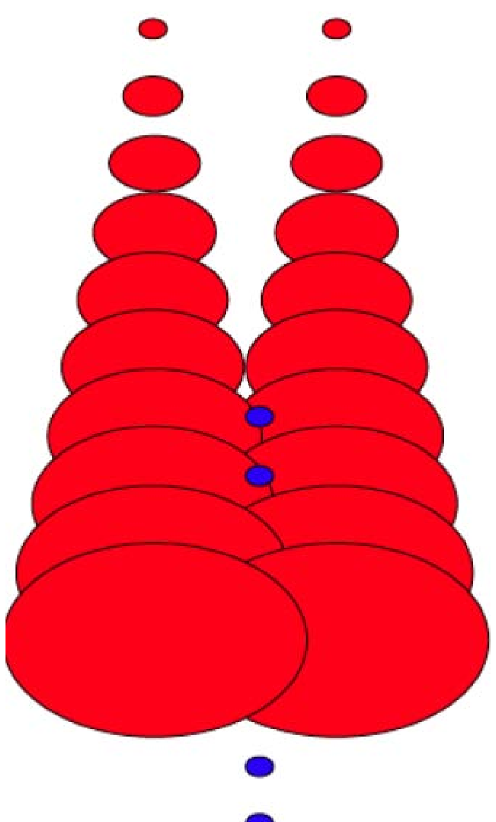
BUT: Severely crowded environments: all paths lower bounded by large cost

Conclusion: *improving prediction* or *reducing covariance* cannot be expected to solve the freezing robot problem

Approaches Continued

"Agnostic" robot:
agent independence

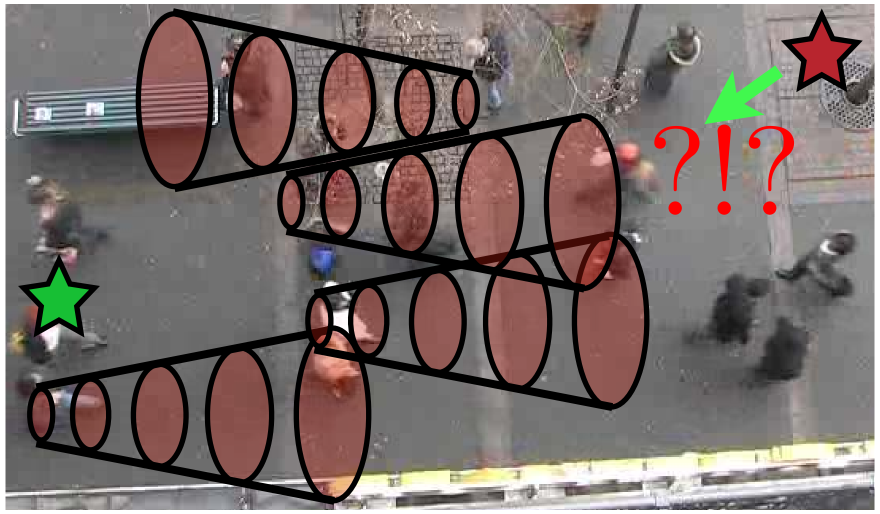


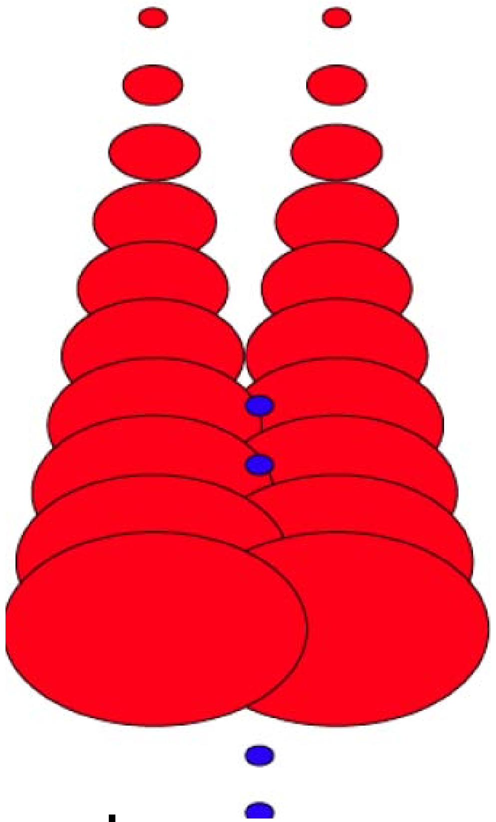


- Const. cov. unreliable
- Reactive planning unreliable

Approaches Continued

"Agnostic" robot: agent independence





"Cooperative" robot: robot/agent system

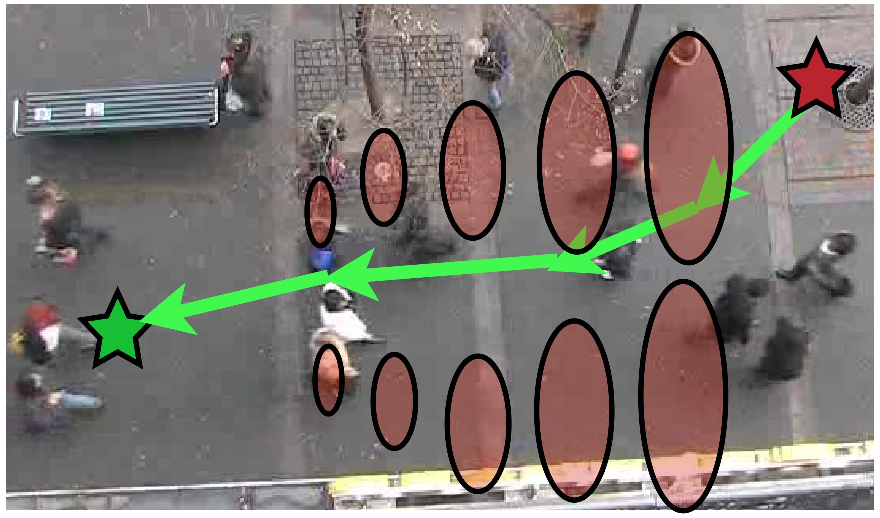
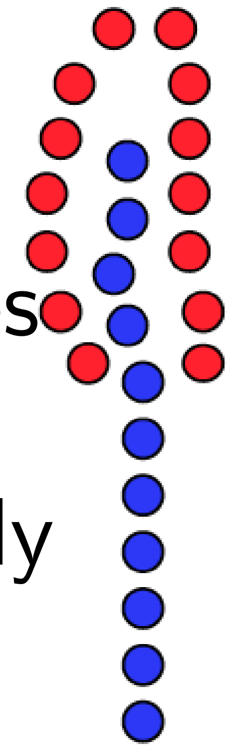
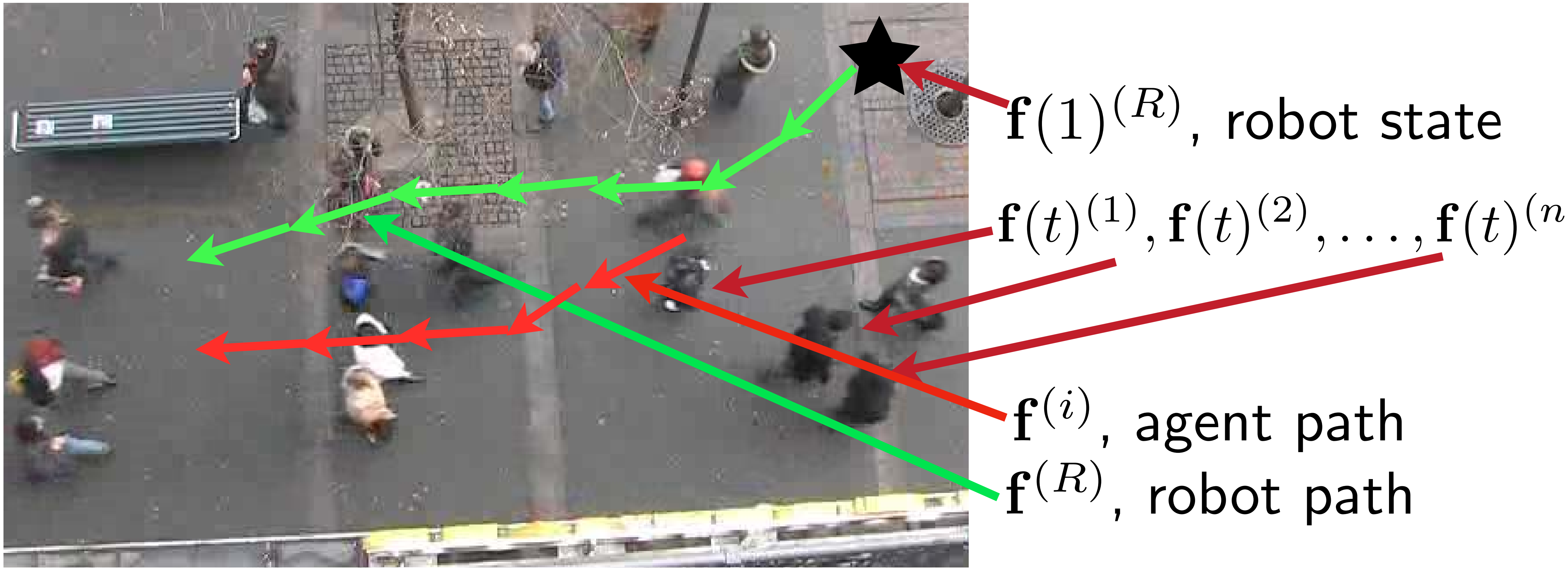


Illustration: robot cooperates with agents to proceed optimally

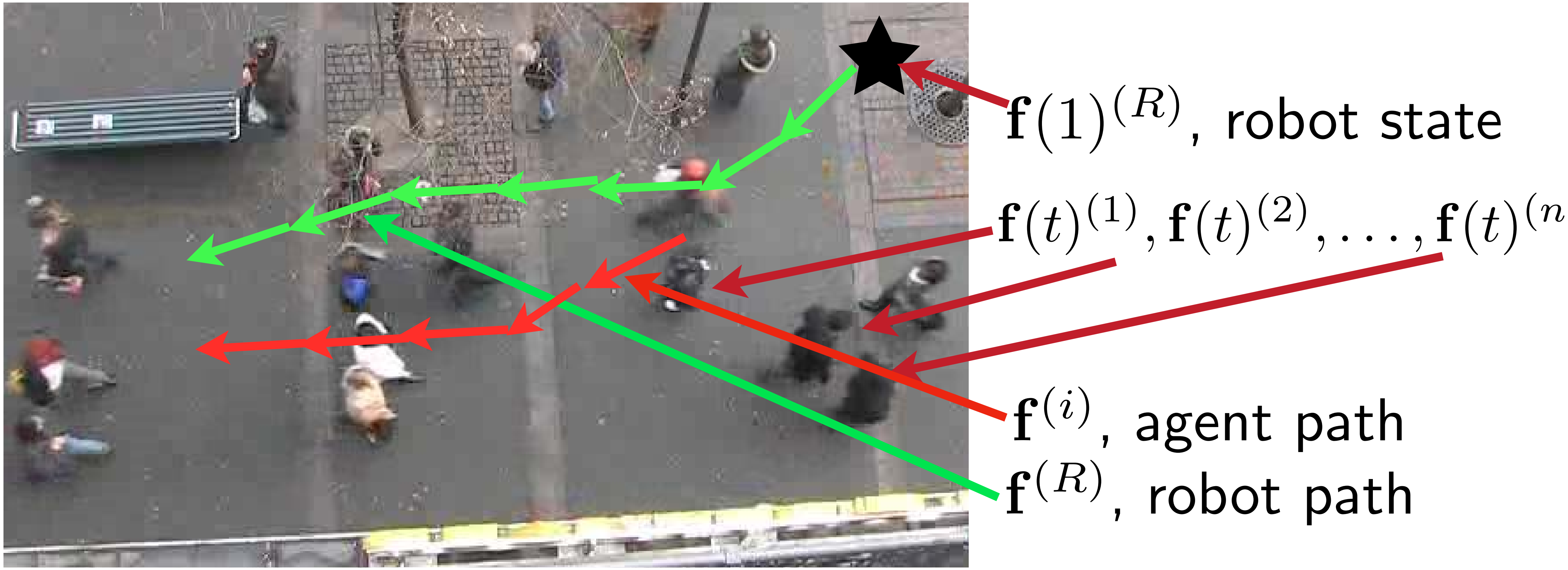


- Const. cov. unreliable
- Reactive planning unreliable

Our Solution



Our Solution



— $\mathbf{f}^{(i)}$ is agent i's *continuous path* in the plane $\mathbb{R}^2$:

$$\mathbf{f}^i \colon [0, \infty] \rightarrow \mathbb{R}^2$$
$$: t \mapsto [x(t)^{(i)}, y(t)^{(i)}]$$

— $\mathbf{f}(t) = (\mathbf{f}(t)^{(1)}, \mathbf{f}(t)^{(2)}, \ldots, \mathbf{f}(t)^{(n)})$ is concatenation of n agent paths

Challenge: how to efficiently and accurately model random, continuous trajectories (functions)

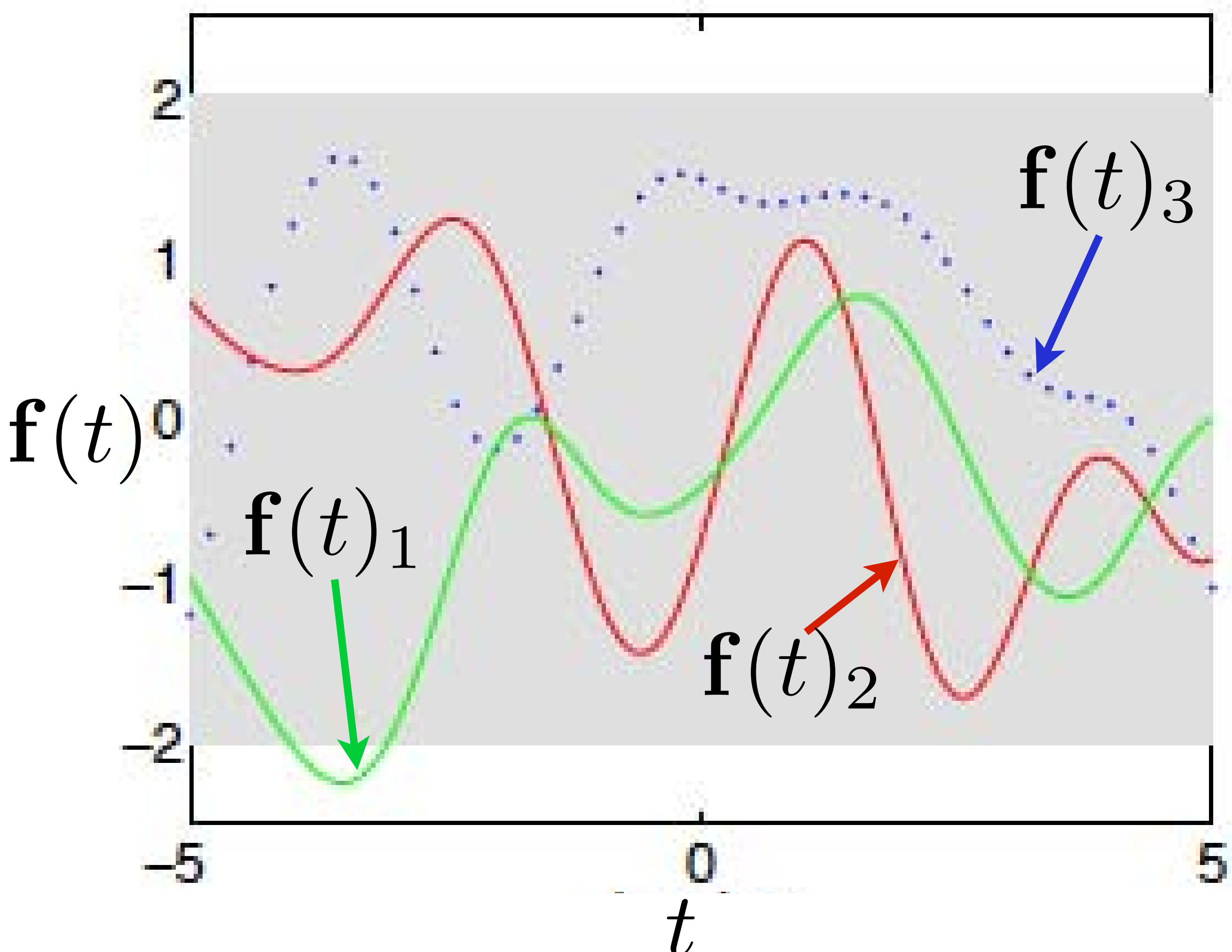
Gaussian Processes for Trajectory Modeling

GP prior: distribution
over functions
$\Longrightarrow$ models trajectories well

Gaussian Processes for Trajectory Modeling

GP prior: distribution over functions

$\Longrightarrow$ models trajectories well



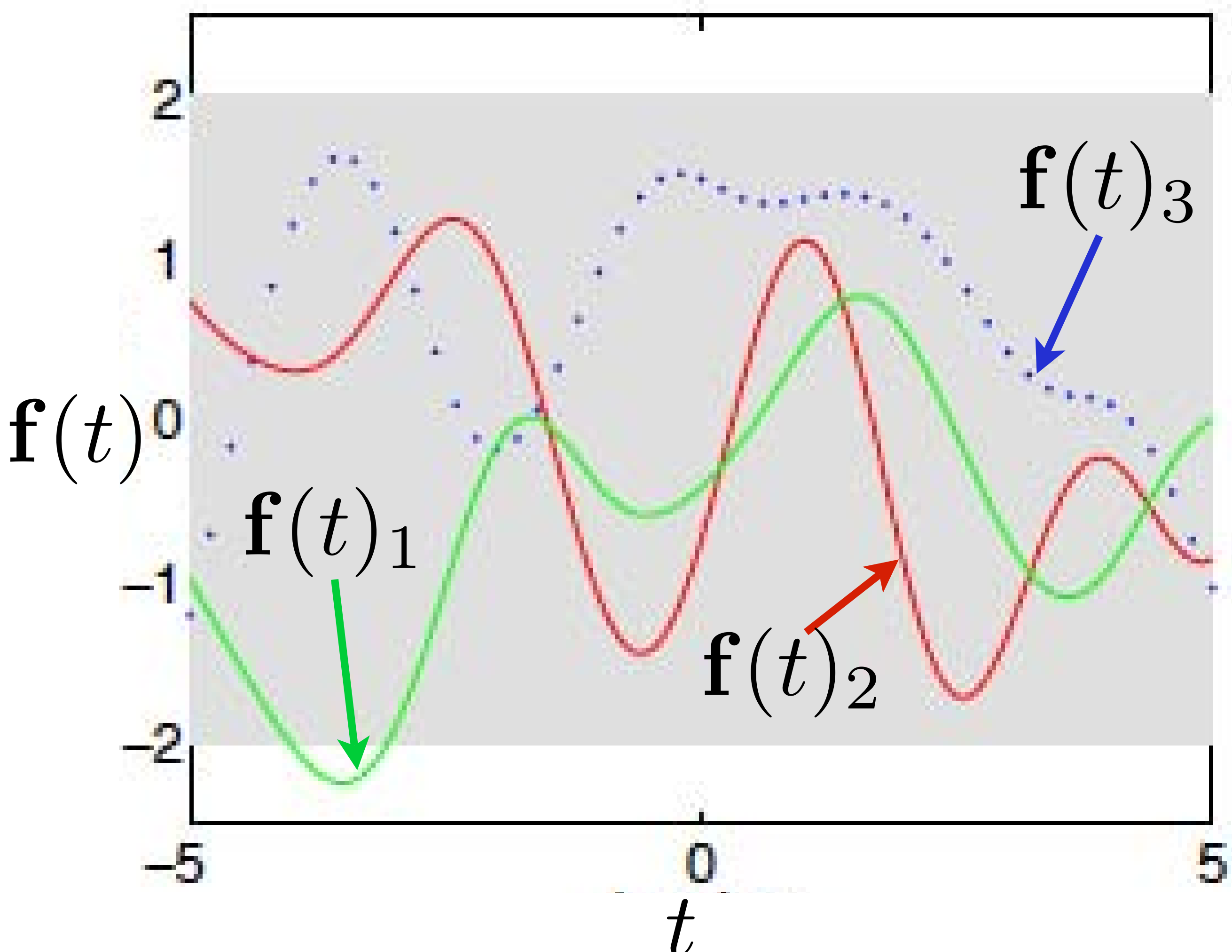
- Draw *trajectories*

  $\mathbf{f}(t)_i \sim GP(0, k)$

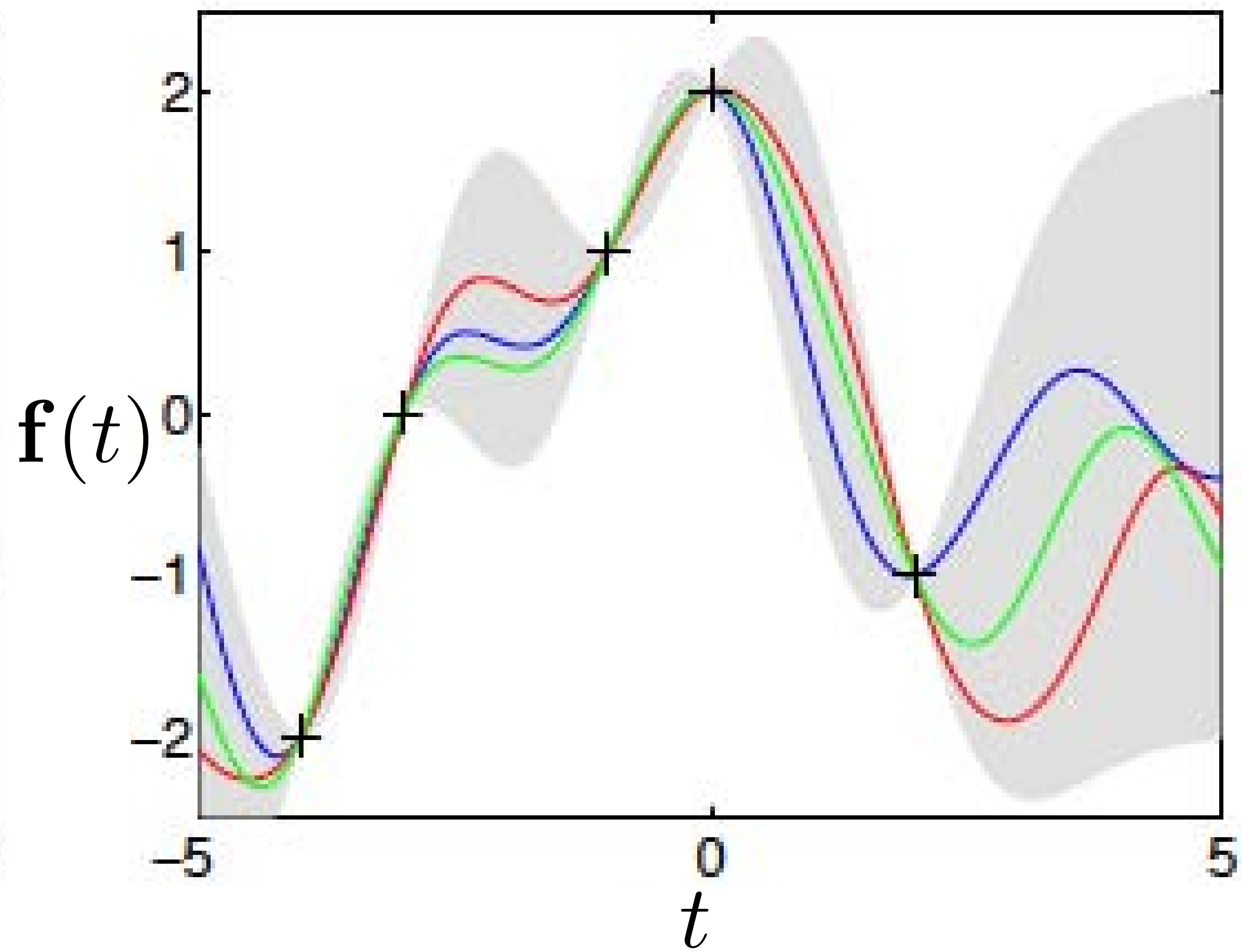
- kernel function k controls smoothness of $x(t)$

Gaussian Processes for Trajectory Modeling

GP prior: distribution over functions $\Longrightarrow$ models trajectories well



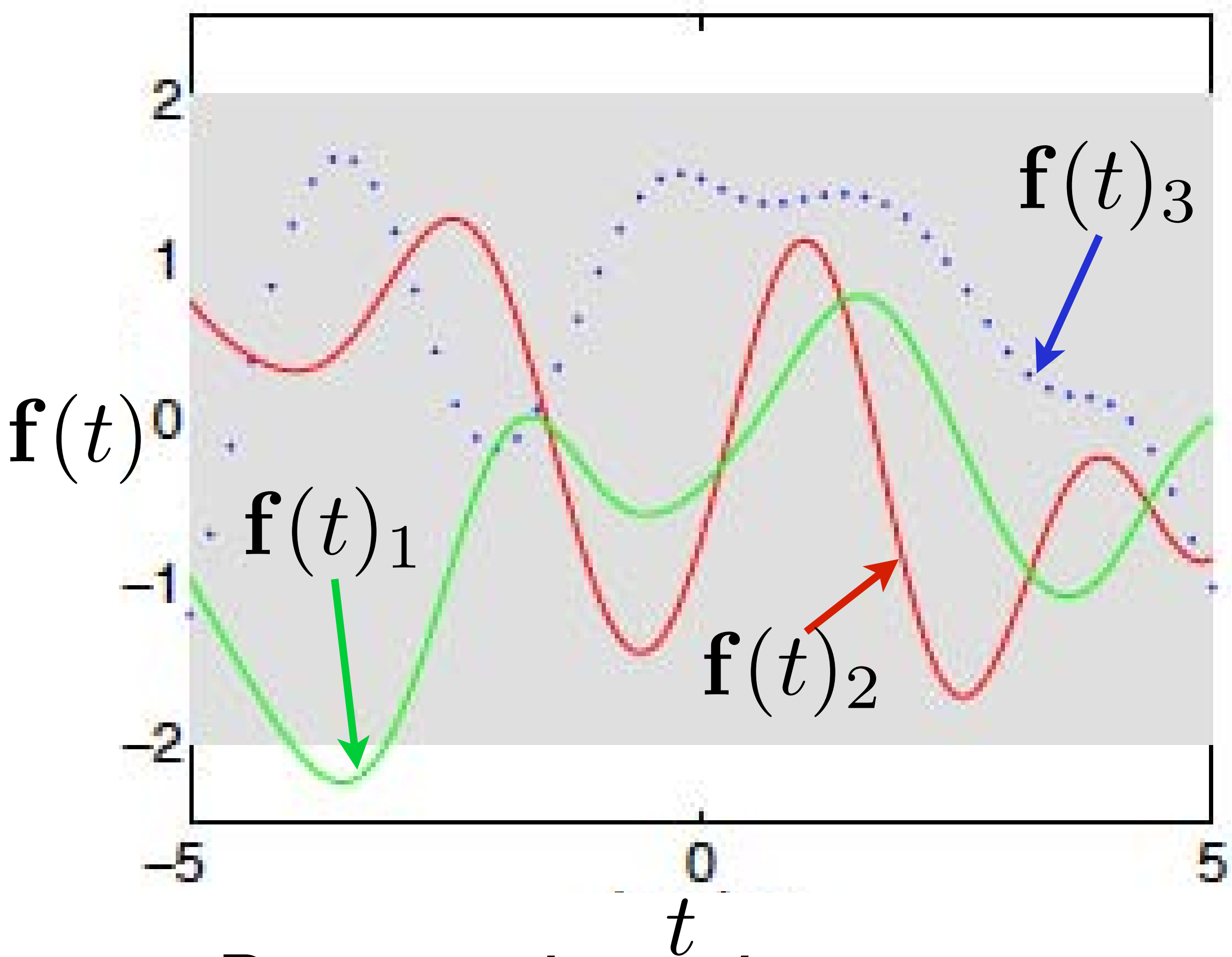
GP posterior: incorporates information at any point along trajectory



- Draw *trajectories*
  $\mathbf{f}(t)_i \sim GP(0, k)$
- kernel function k controls smoothness of $x(t)$

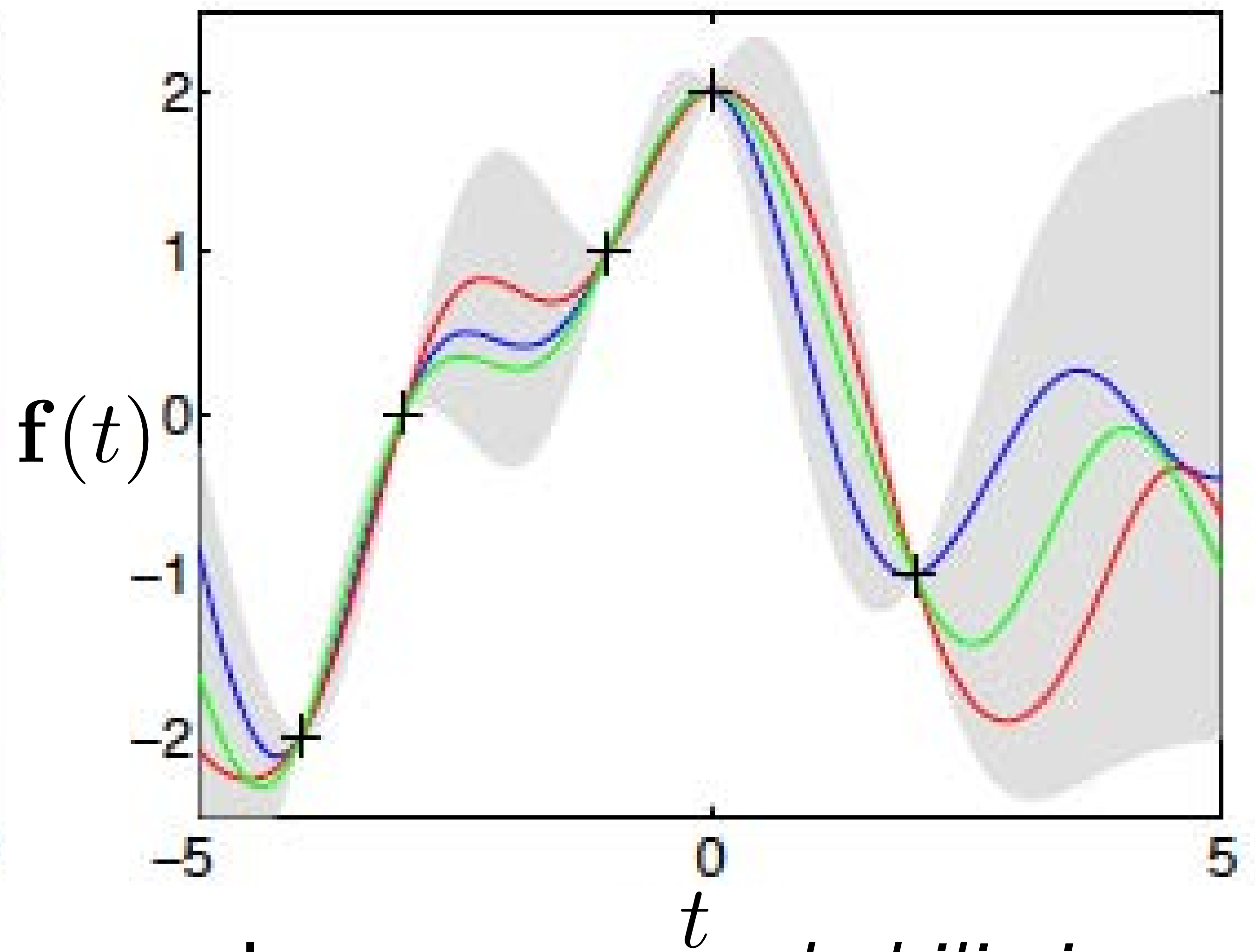
Gaussian Processes for Trajectory Modeling

GP prior: distribution over functions
$\Longrightarrow$ models trajectories well



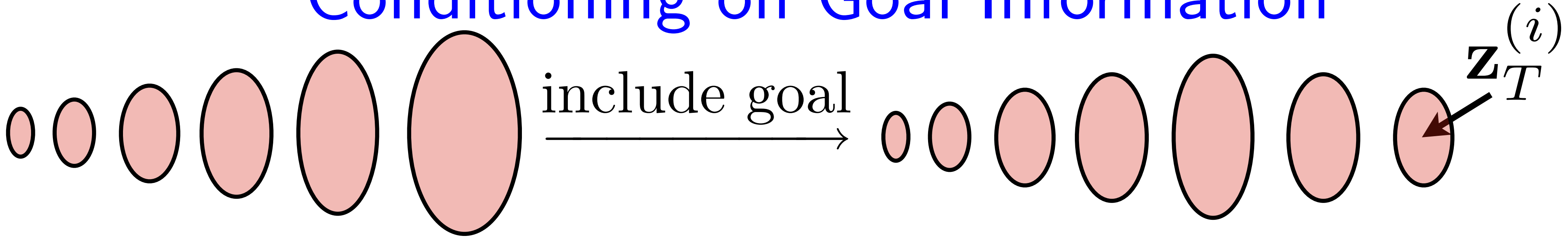
- Draw *trajectories*
  $\mathbf{f}(t)_i \sim GP(0, k)$
- kernel function k controls smoothness of $x(t)$

GP posterior: incorporates information at any point along trajectory

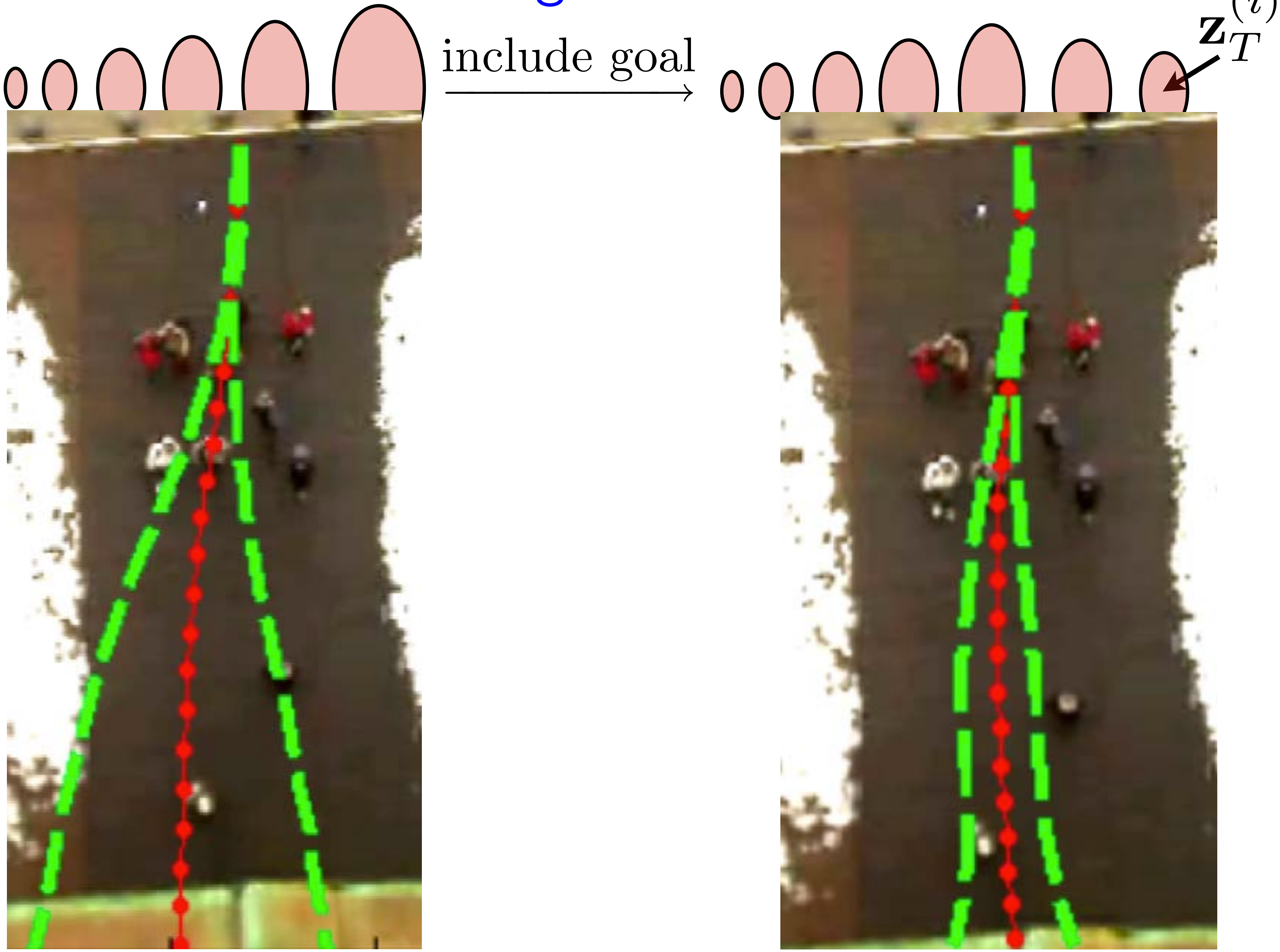


- Incorporate *probabilistic* goal information
- Encode smoothness in a non-Markovian way

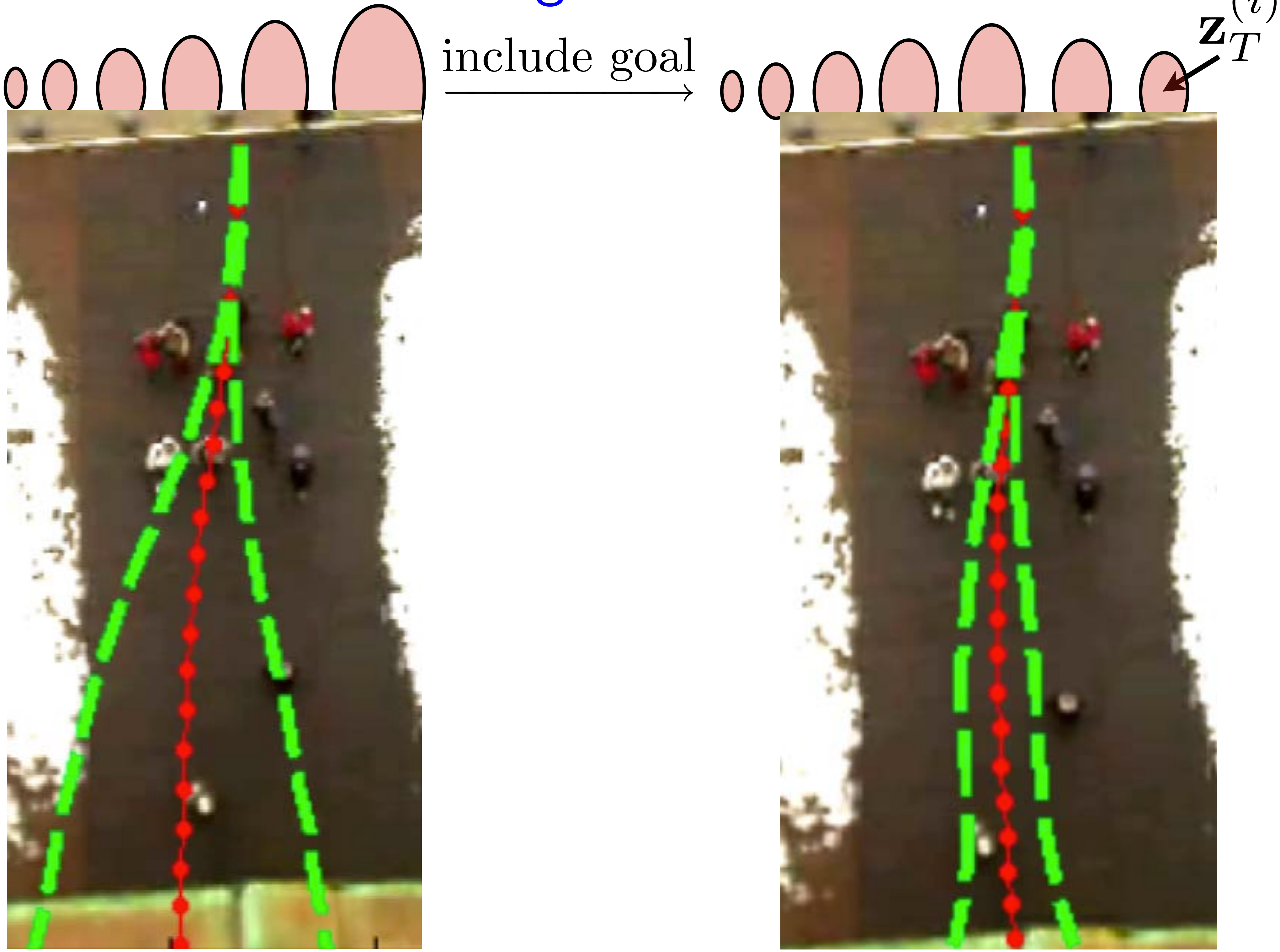
Conditioning on Goal Information



Conditioning on Goal Information



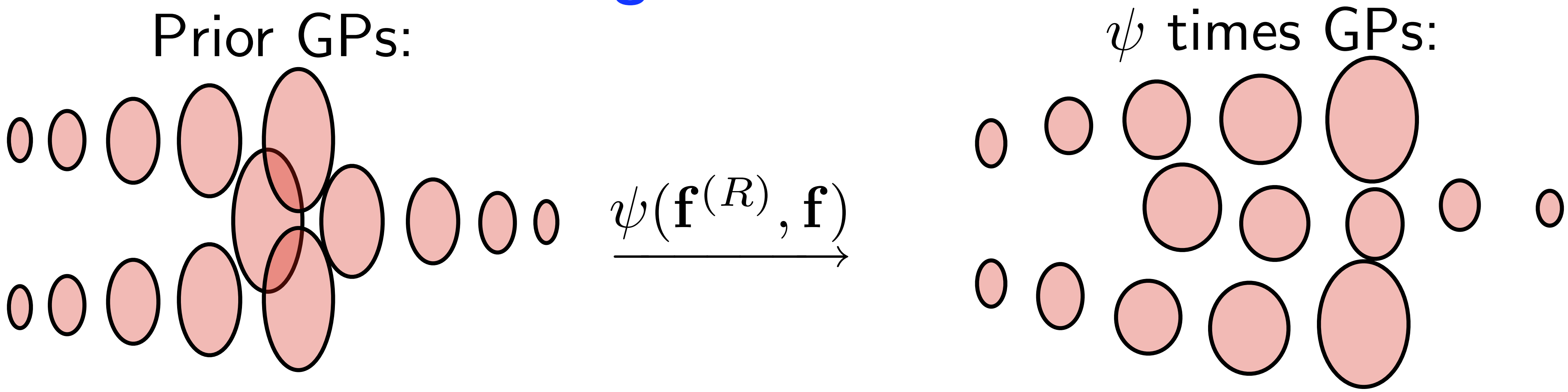
Conditioning on Goal Information



Train prior GPs to find $k^{(i)}$, condition on goal information $\mathbf{z}_T^{(i)}$:

$$p_{k^{(R)}}(\mathbf{f}^{(R)} \mid \mathbf{z}_{1:t}, \mathbf{z}_T^{(R)}), p_{k^{(1)}}(\mathbf{f}^{(1)} \mid \mathbf{z}_{1:t}^{(1)}, \mathbf{z}_T^{(1)}), \ldots, p_{k^{(n)}}(\mathbf{f}^{(n)} \mid \mathbf{z}_{1:t}, \mathbf{z}_T^{(n)})$$

Interacting Gaussian Processes



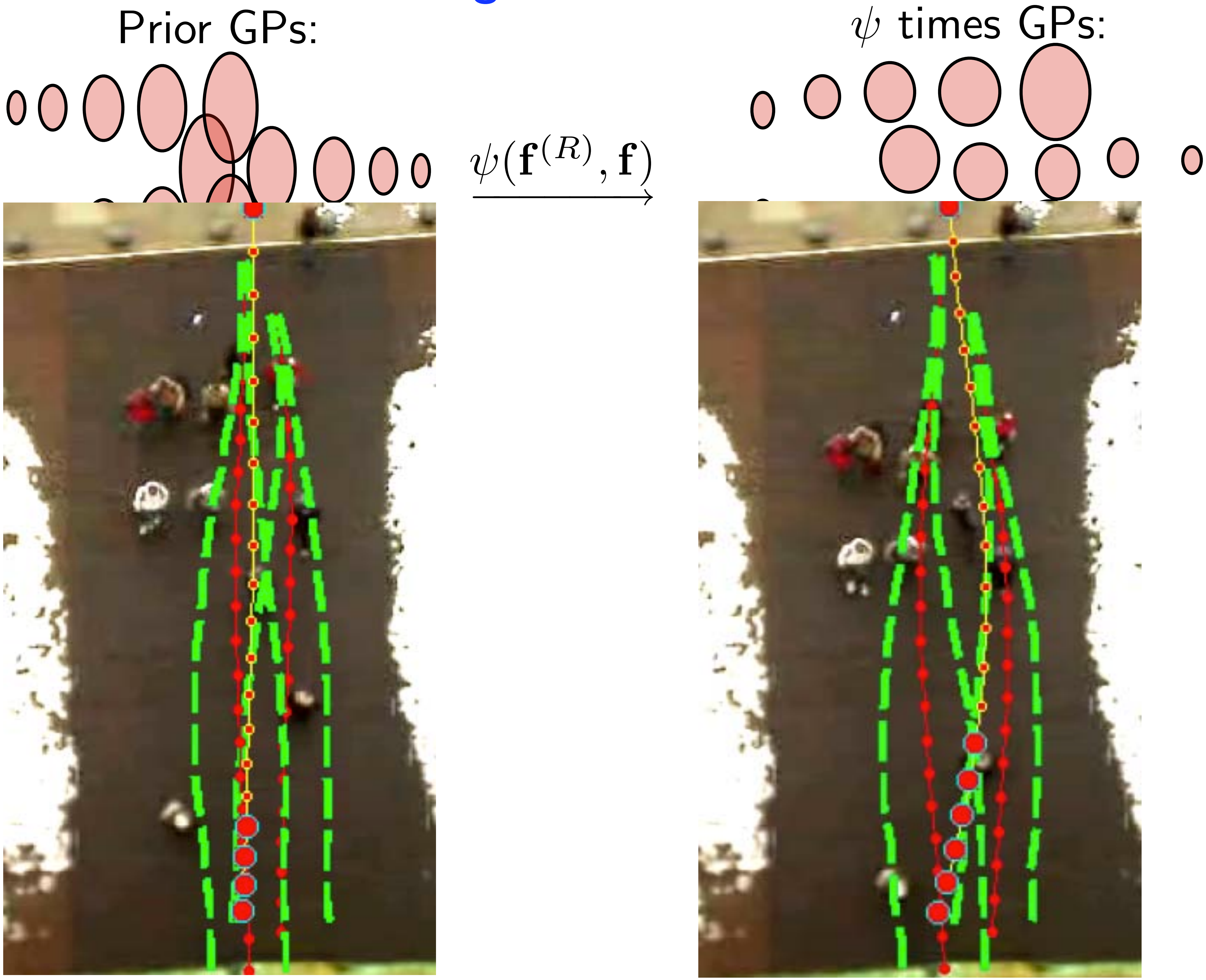
Interacting Gaussian Processes

Prior GPs:

$$\xrightarrow{\psi(\mathbf{f}^{(R)}, \mathbf{f})}$$

ψ times GPs:

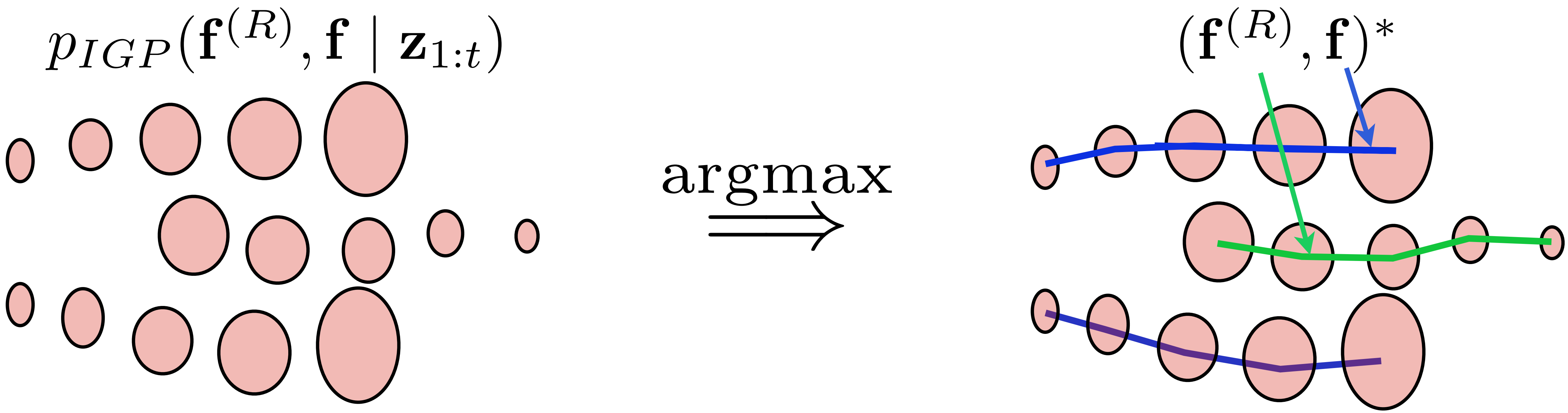
Interacting Gaussian Processes



$$p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t}) = \frac{1}{Z}\psi(\mathbf{f}^{(R)}, \mathbf{f}) \prod_{i=R}^{n} p_{k^{(i)}}(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}, \mathbf{z}_T^{(i)})$$

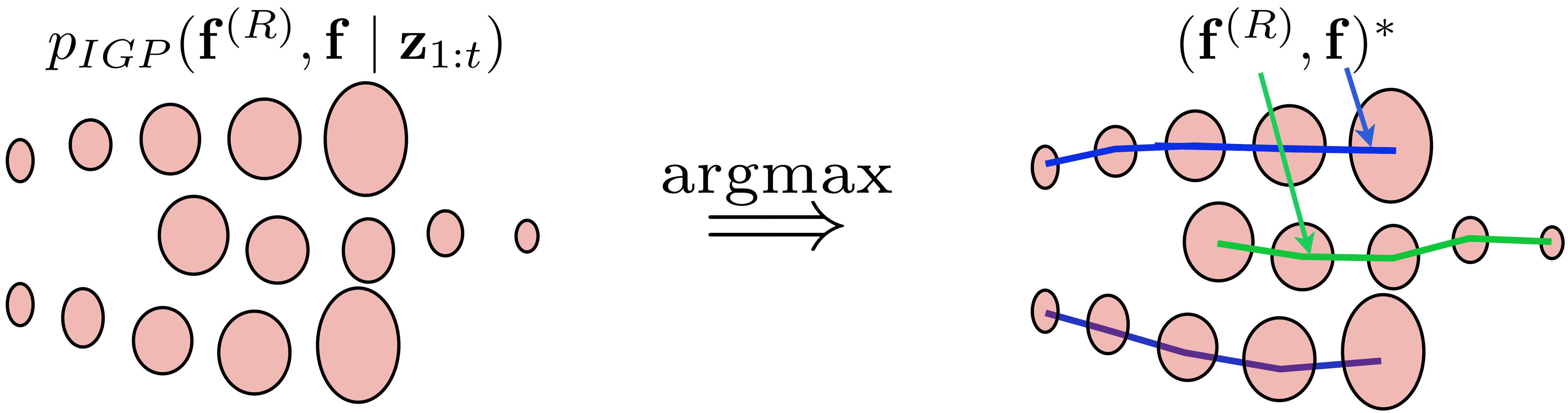
Navigation with Interacting Gaussian Processes

$p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t})$ suggests a natural way to perform navigation: at time t, find that MAP of the posterior



Navigation with Interacting Gaussian Processes

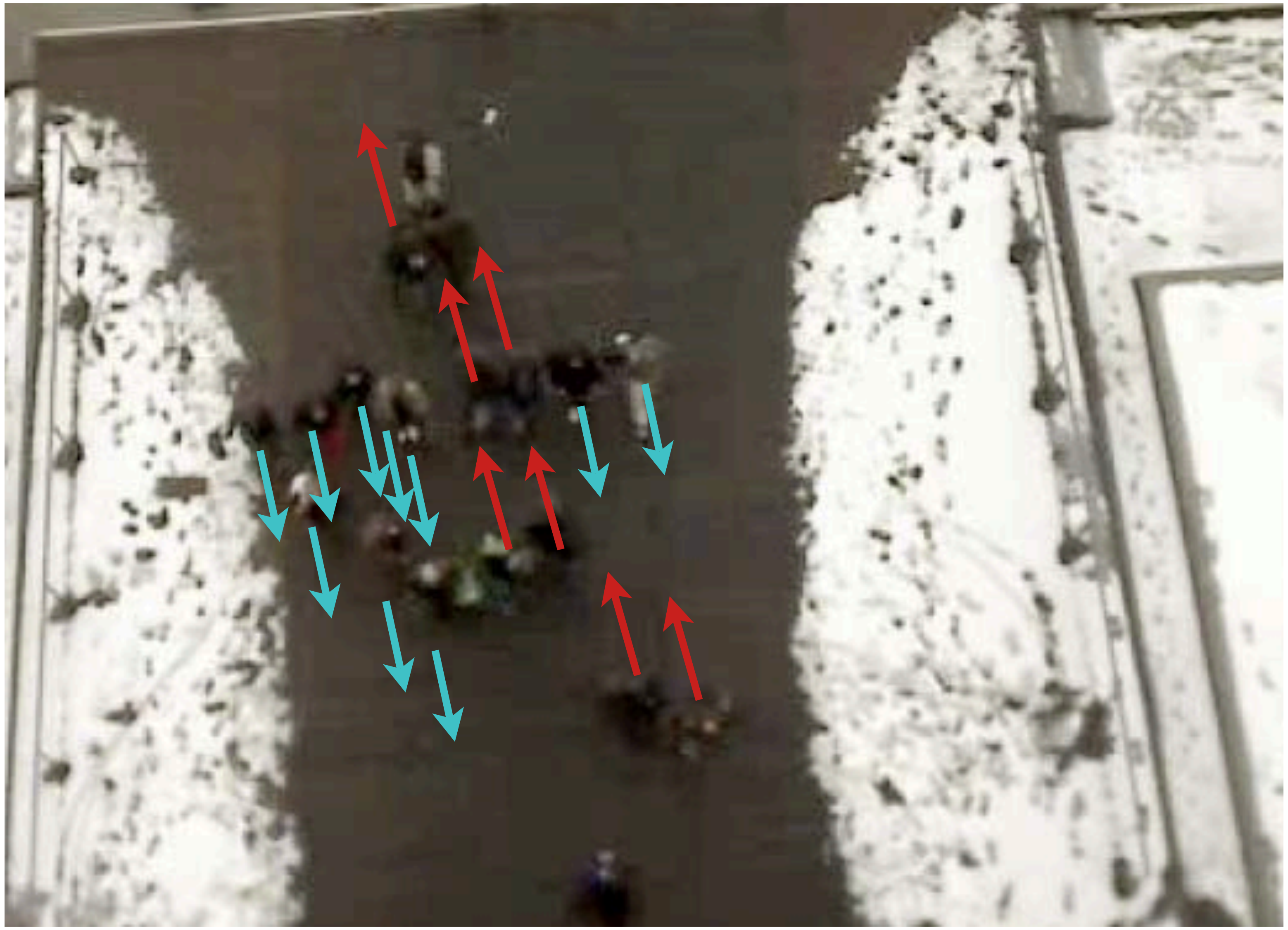
$p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t})$ suggests a natural way to perform navigation: at time t, find that MAP of the posterior



$$(\mathbf{f}^{(R)}, \mathbf{f})^* = \arg\max_{\mathbf{f}^{(R)}, \mathbf{f}} p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t})$$

and take $f^{(R)}(t+1)^*$ as the next action in the path

Evaluation: ETH Data Set



- Very dense crowds, large numbers of pedestrians
- Many instances of crowd interaction
- Goal oriented behavior
- Ground truthed for over 8 minutes

Results

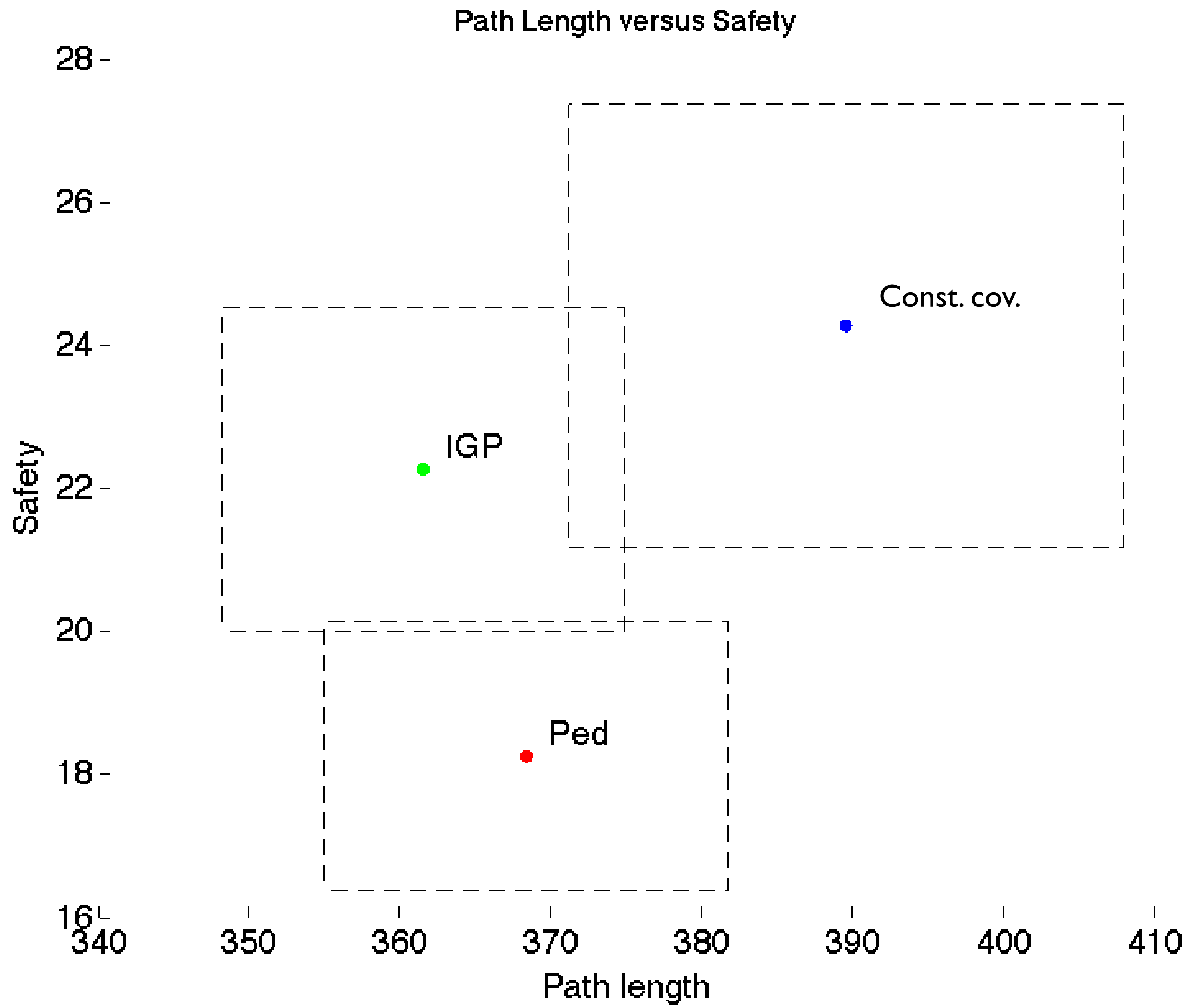
Const. covariance *IGP and pedestrian* *IGP and pedestrian*

Const. cov. evasive; IGP similar to pedestrian

Results

Path length vs safety, 10 runs
IGP outperforms pedestrians.
Constant cov inappropriate

Results



Path length vs safety, 10 runs

IGP outperforms pedestrians.

Constant cov inappropriate

Runtime in Matlab: $\approx$ 10Hz

Conclusions

- Defined frozen robot problem
- In dense crowds, freezing robot problem occurs if $p(\mathbf{f} \mid \mathbf{z}_{1:t}) = \prod_{i=1}^{n} p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t})$ assumed
- Navigation in dense crowds requires interaction modeling
- IGP, a nonparametric statistical model based on dependent output GPs
- Navigation $=$'s inference in this model
- Evaluation on real world pedestrian data
- IGP outperformed pedestrians, state of the art algorithm
- Future: develop experiments for cafeteria at lunchtime