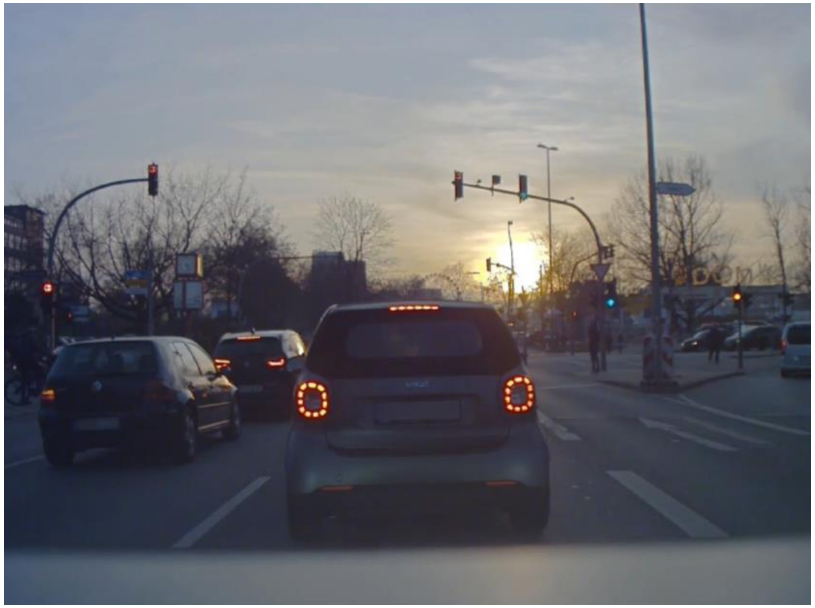
Self-Driving Cars: Localization

Jens Behley, Daniel Wilbers

University of Bonn

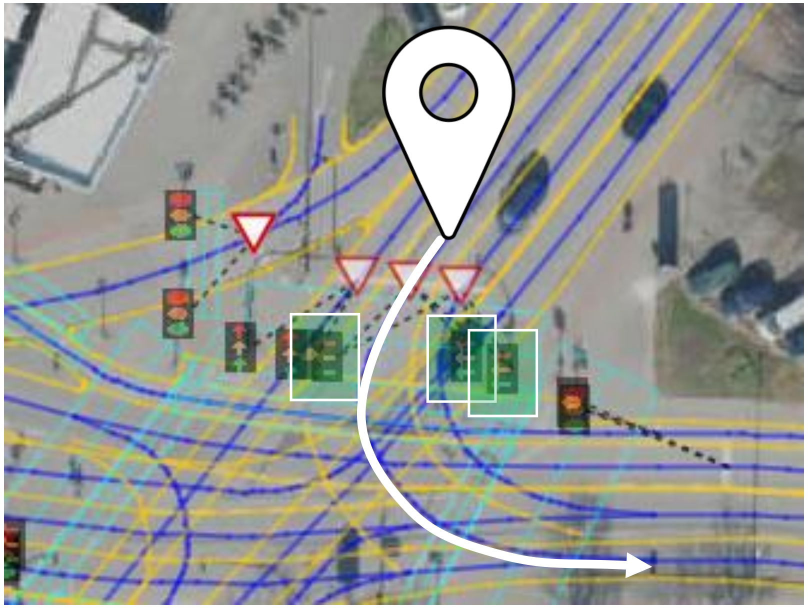
Localization for automated driving

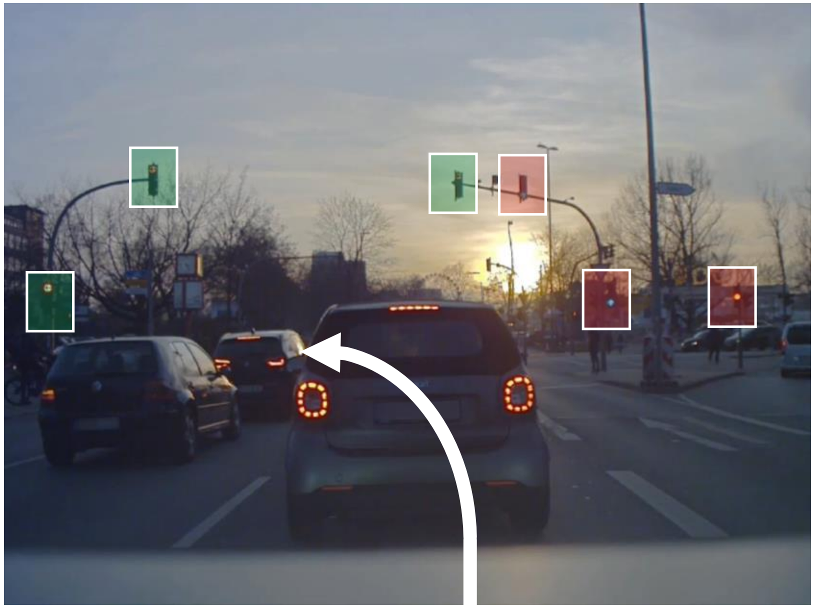



Localization enriches perception and eases scene understanding and planning tasks

[Images courtesy of Volkswagen AG]

Localization for automated driving

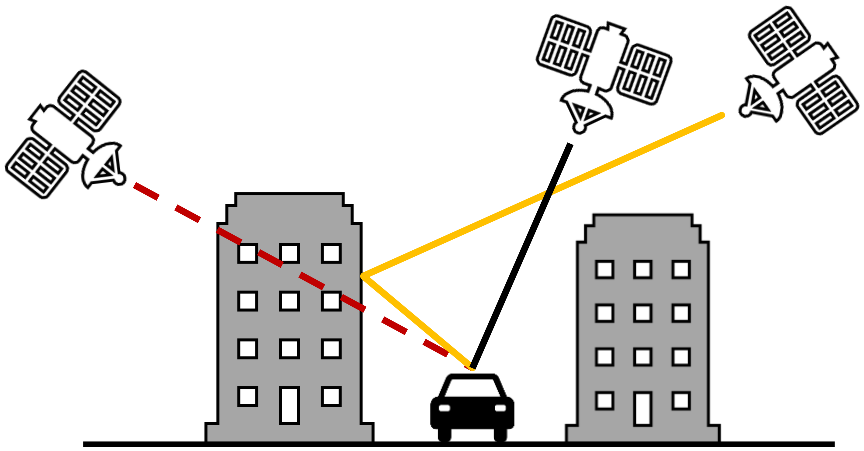


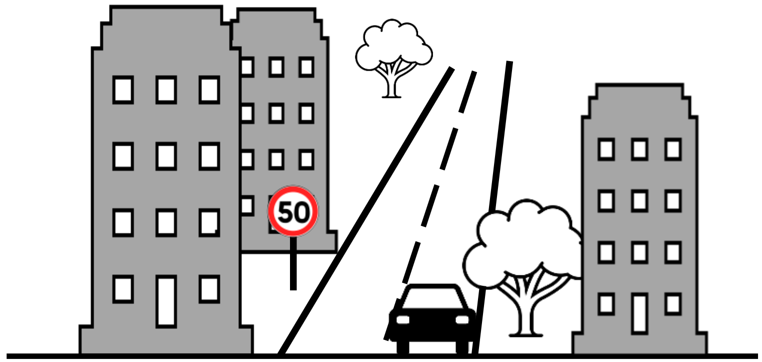


Highly accurate localization enables other automated driving functions

[Images courtesy of Volkswagen AG]

Localization techniques





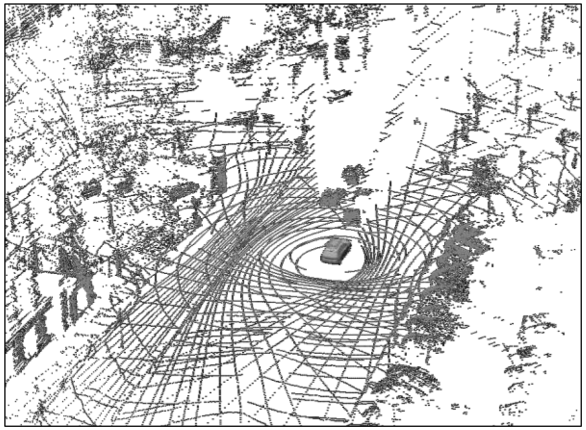
- GNSS-based
- environment perception-based

Perception-based localization

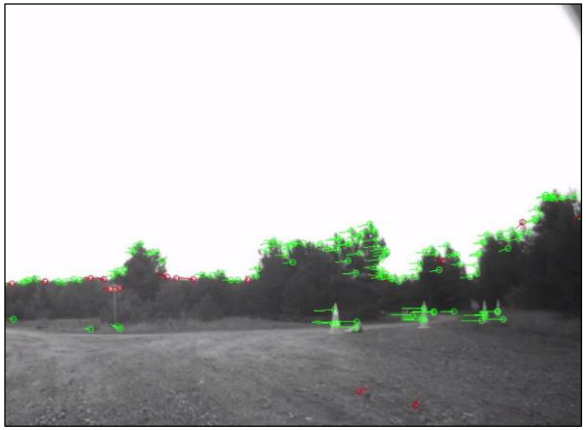
sensor-specific vs. multimodal

[Image courtesy of Volkswagen AG]

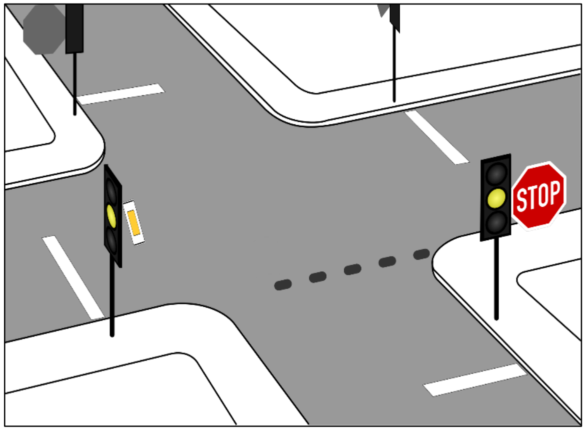
Patterns for localization



Raw data



Features

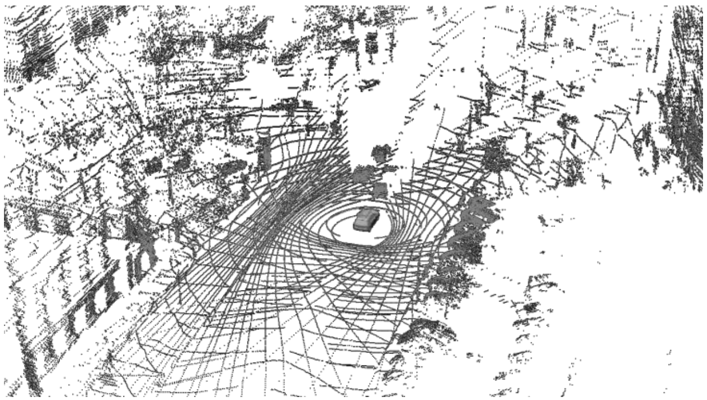


Landmarks

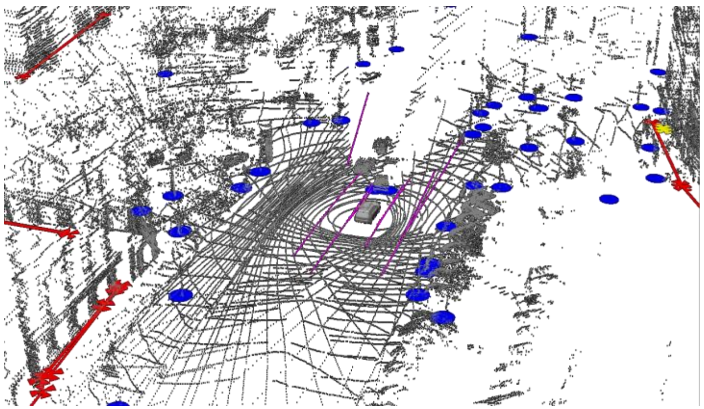
- choice depends on application and environment

[Images courtesy of Volkswagen AG]

Landmarks for urban localization



raw data (e.g. point cloud)

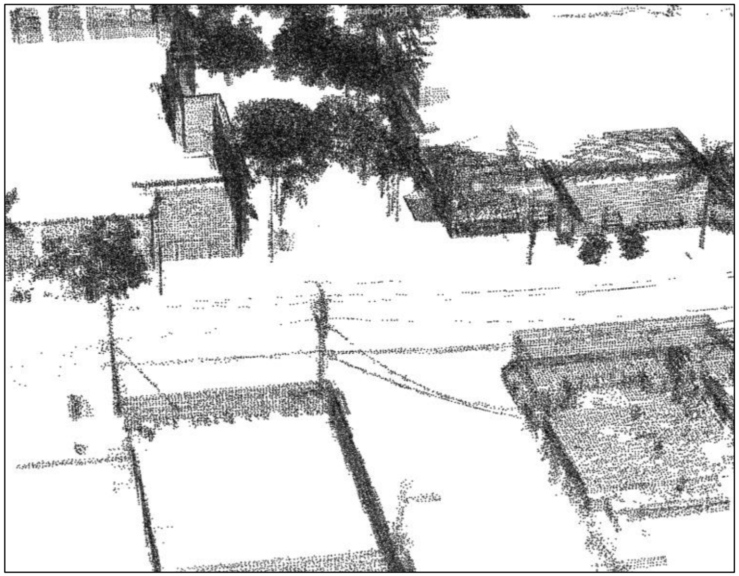


landmark detection

- Pros: storage space, multi-sensor, computation time
- Cons: depends on available landmarks, map must be up to date

[Images courtesy of Volkswagen AG]

Maps for localization



LiDAR points

Landmarks:

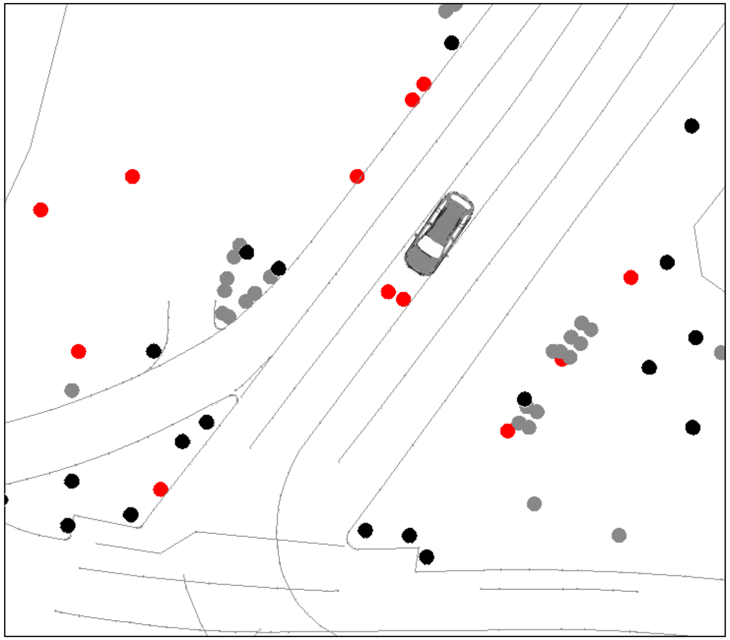
poles building planes

curbs road markings

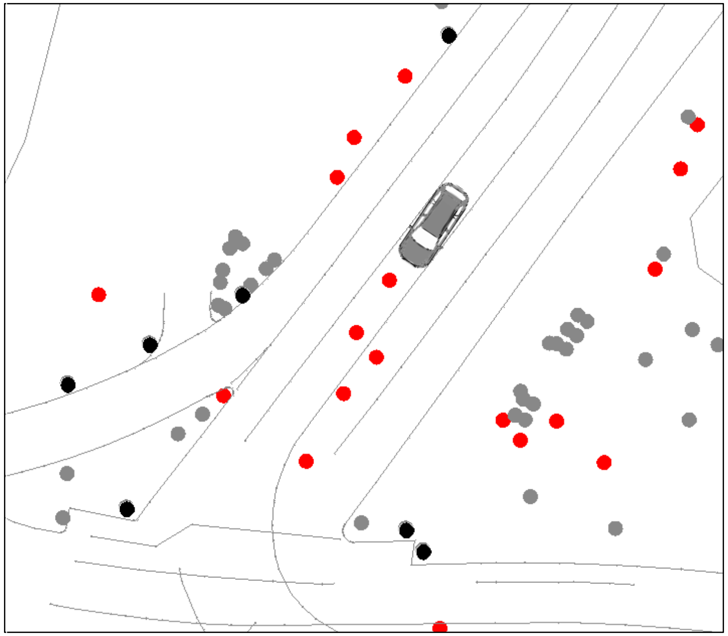
tailored vs. general-purpose maps

[Images courtesy of Volkswagen AG]

Using general-purpose maps



LiDAR landmarks



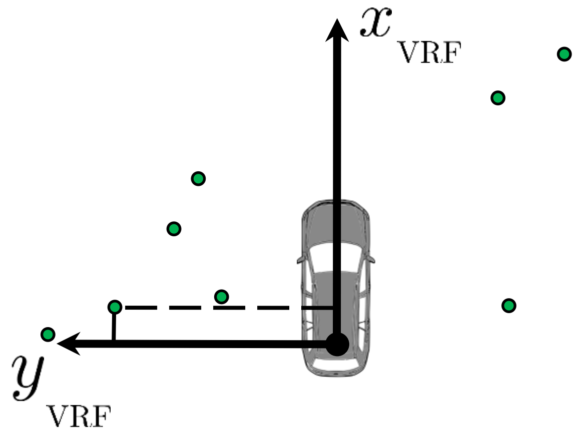
Radar landmarks

- general-purpose map
- match
- false positive or missing in the map

Landmark-based localization

Given

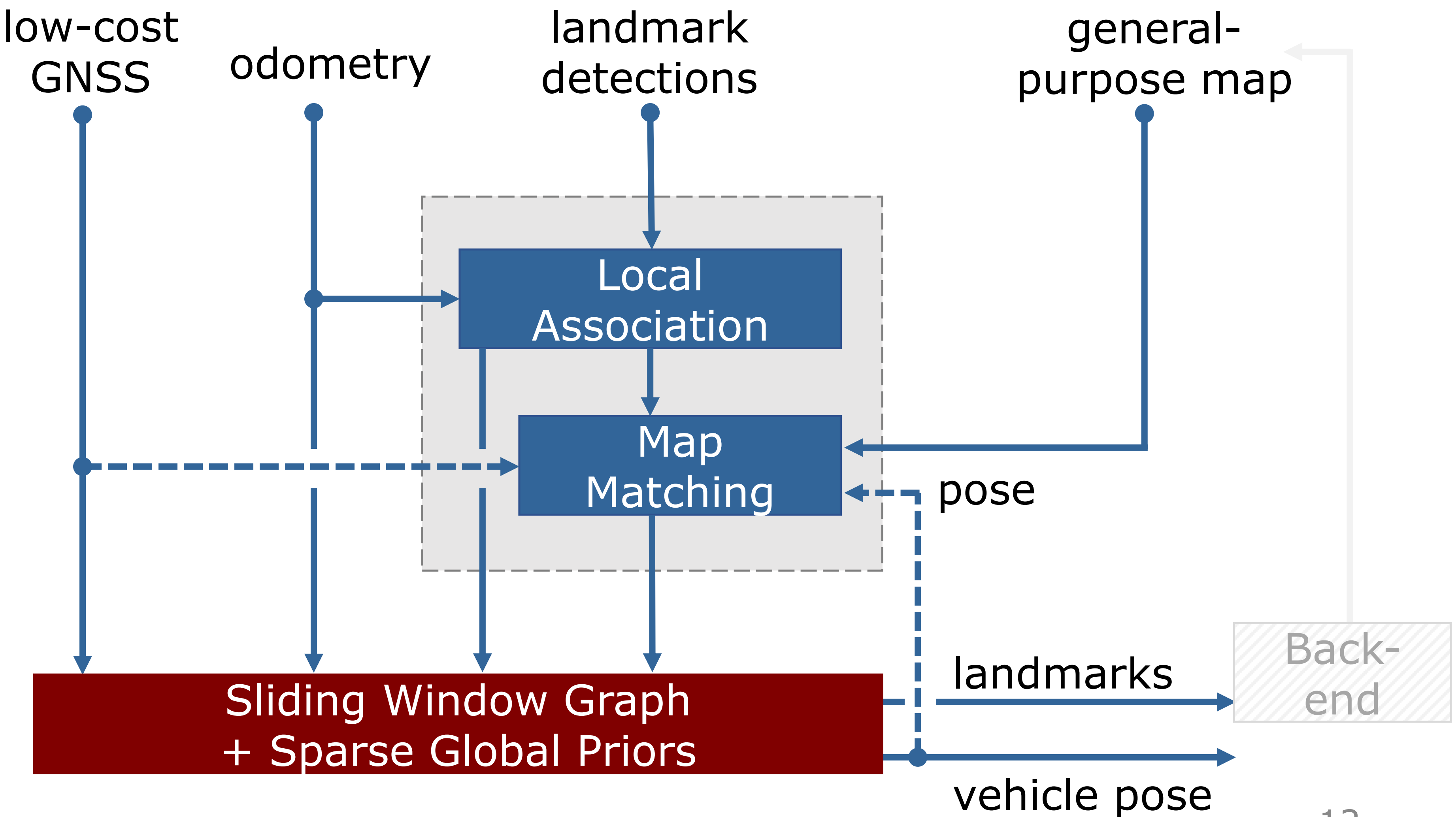
- landmark detections
- odometry
- general-purpose map
- low-cost GNSS

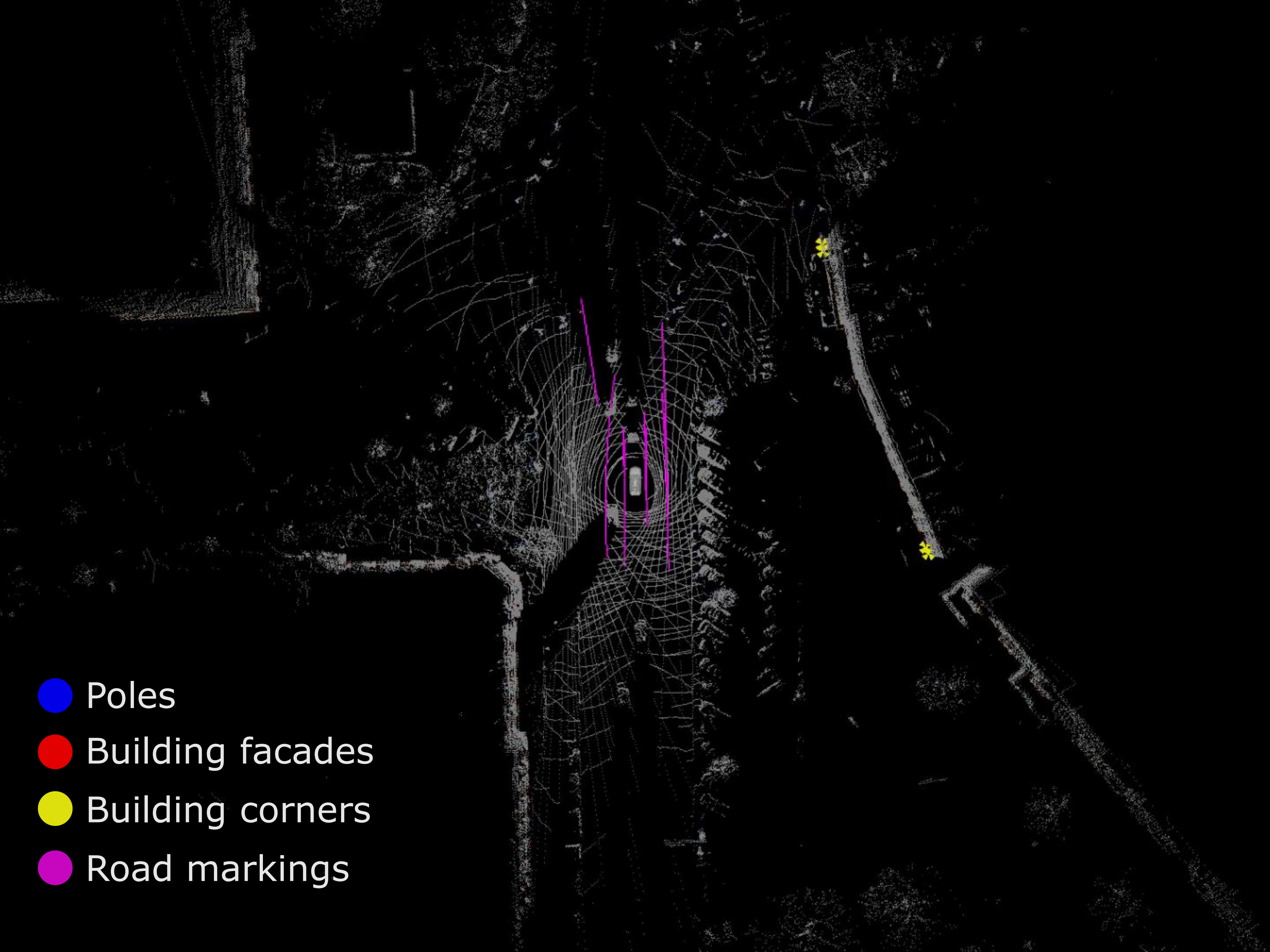


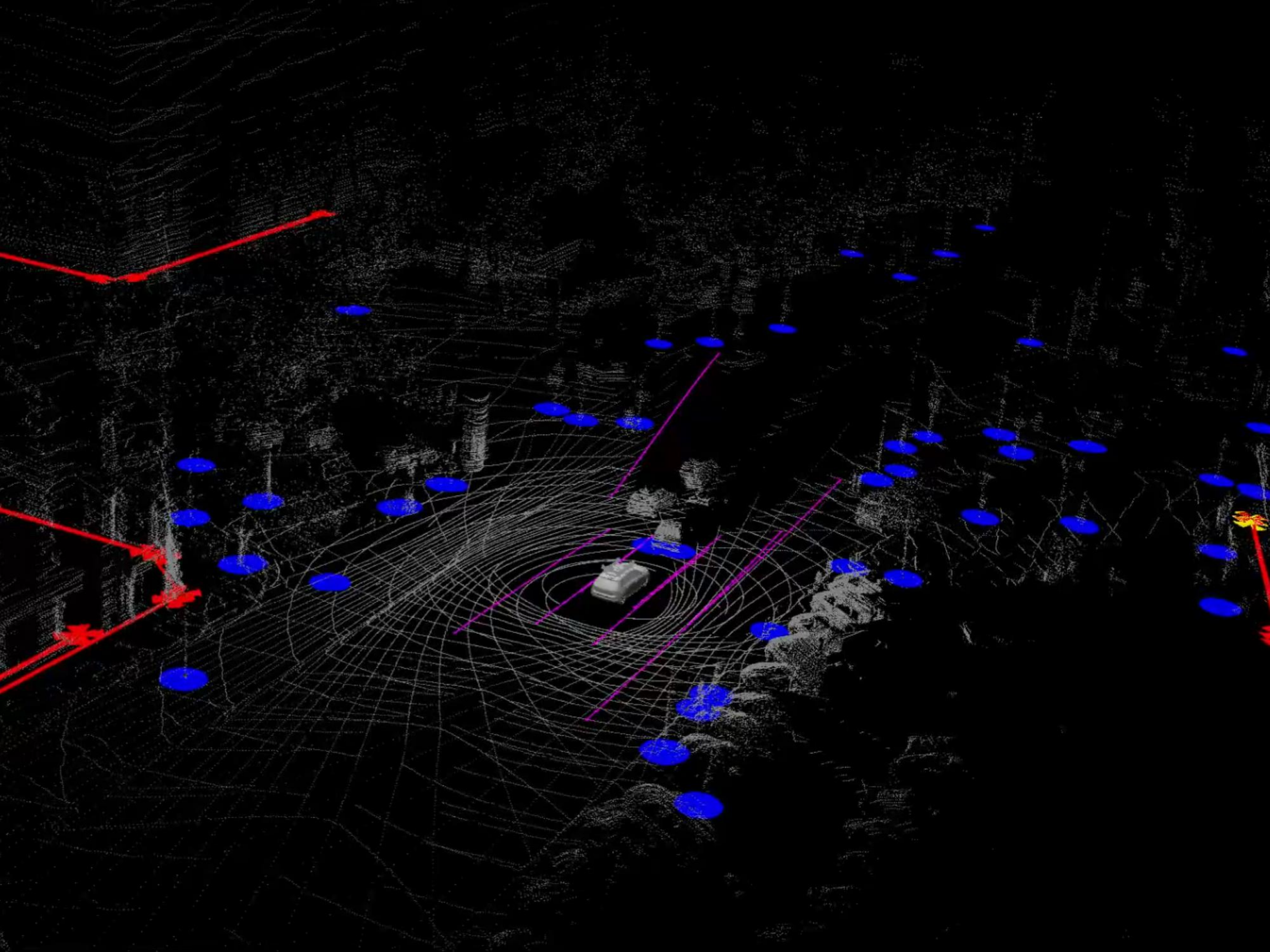
Find

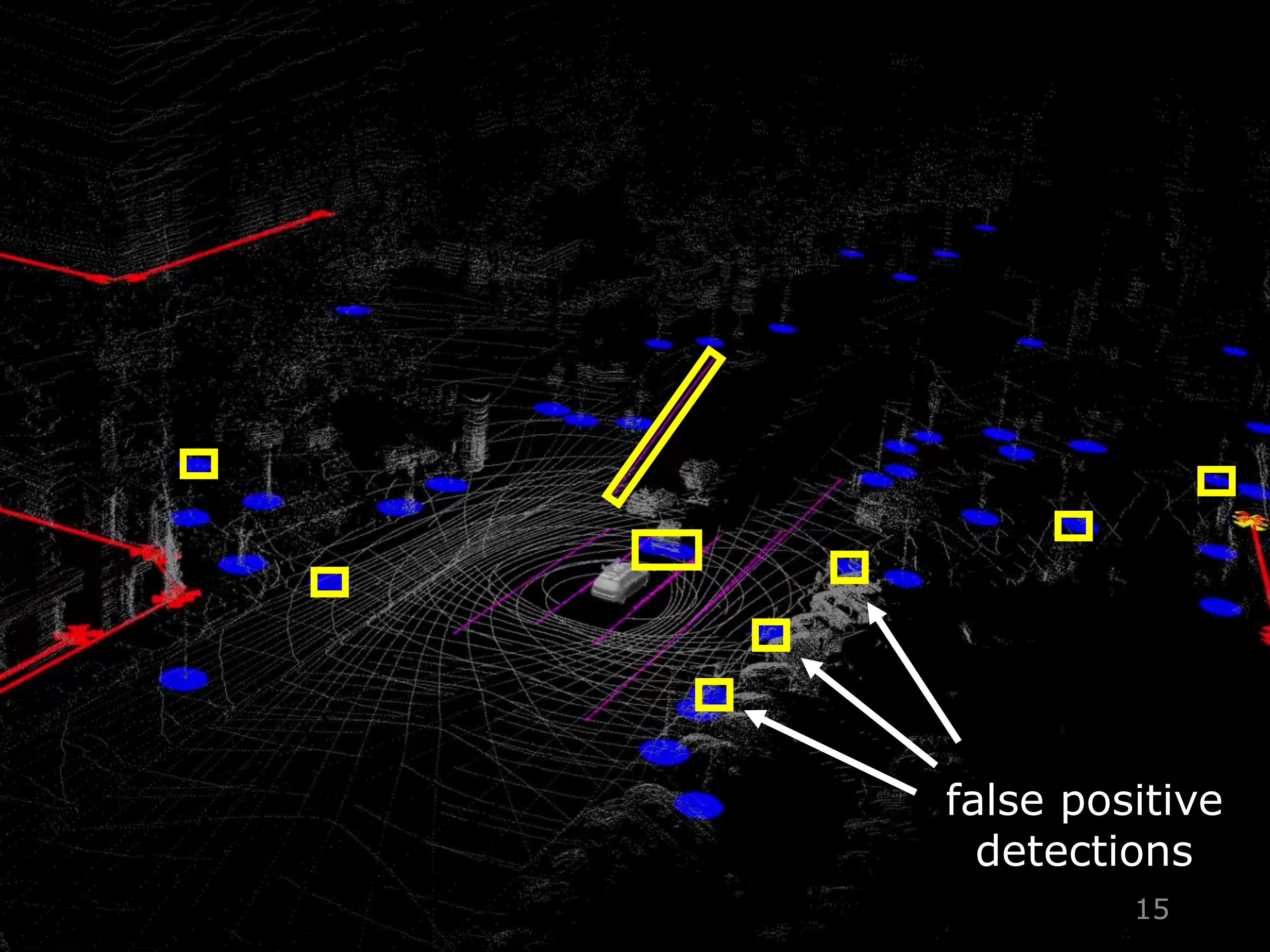
- vehicle trajectory
- landmark positions

Algorithm (simplified)



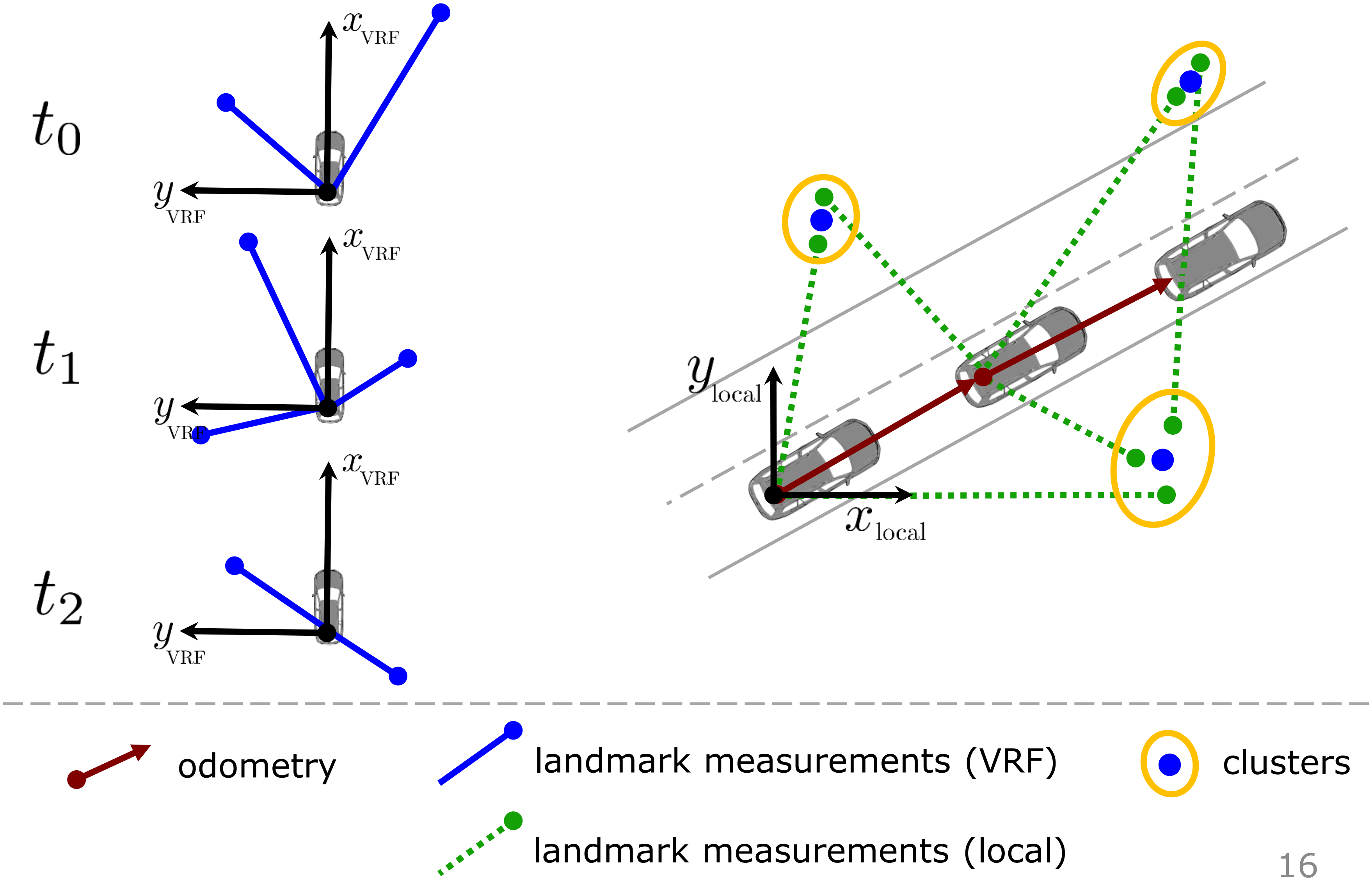

Poles
Building facades
Building corners
Road markings



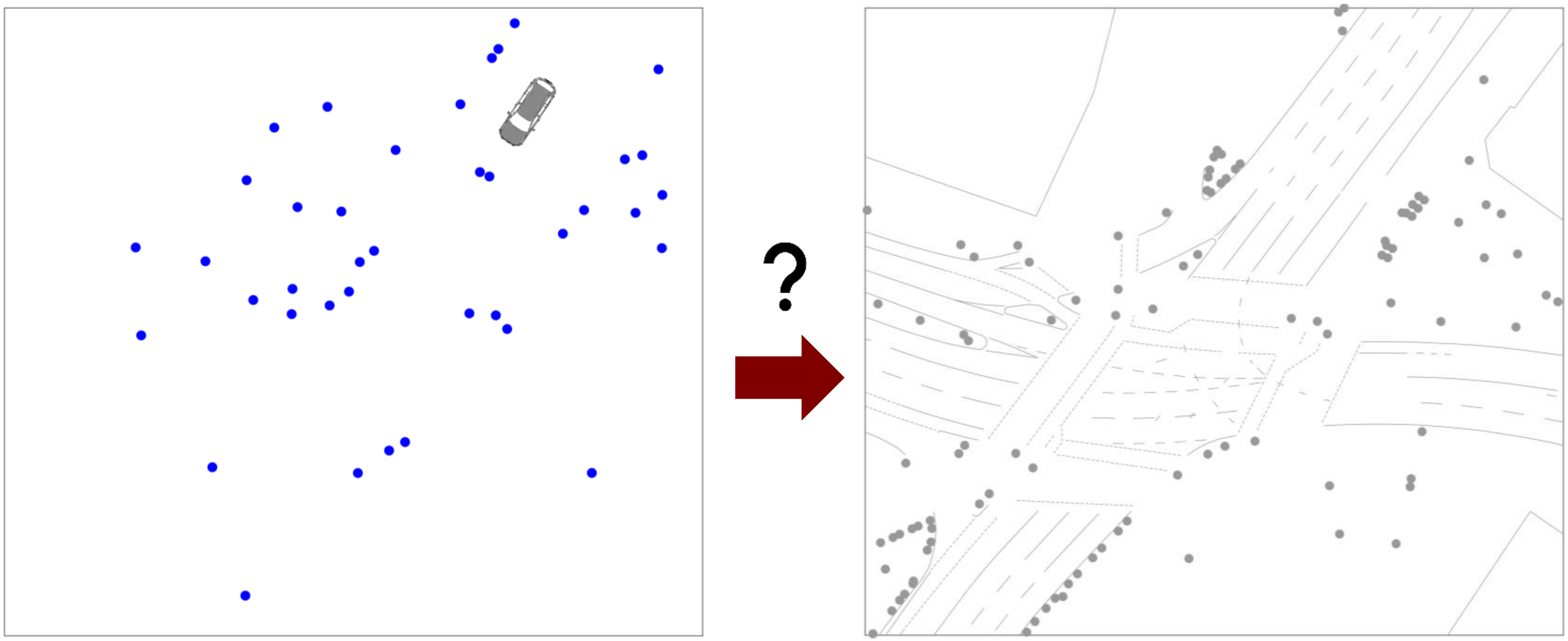


false positive
detections

Local association

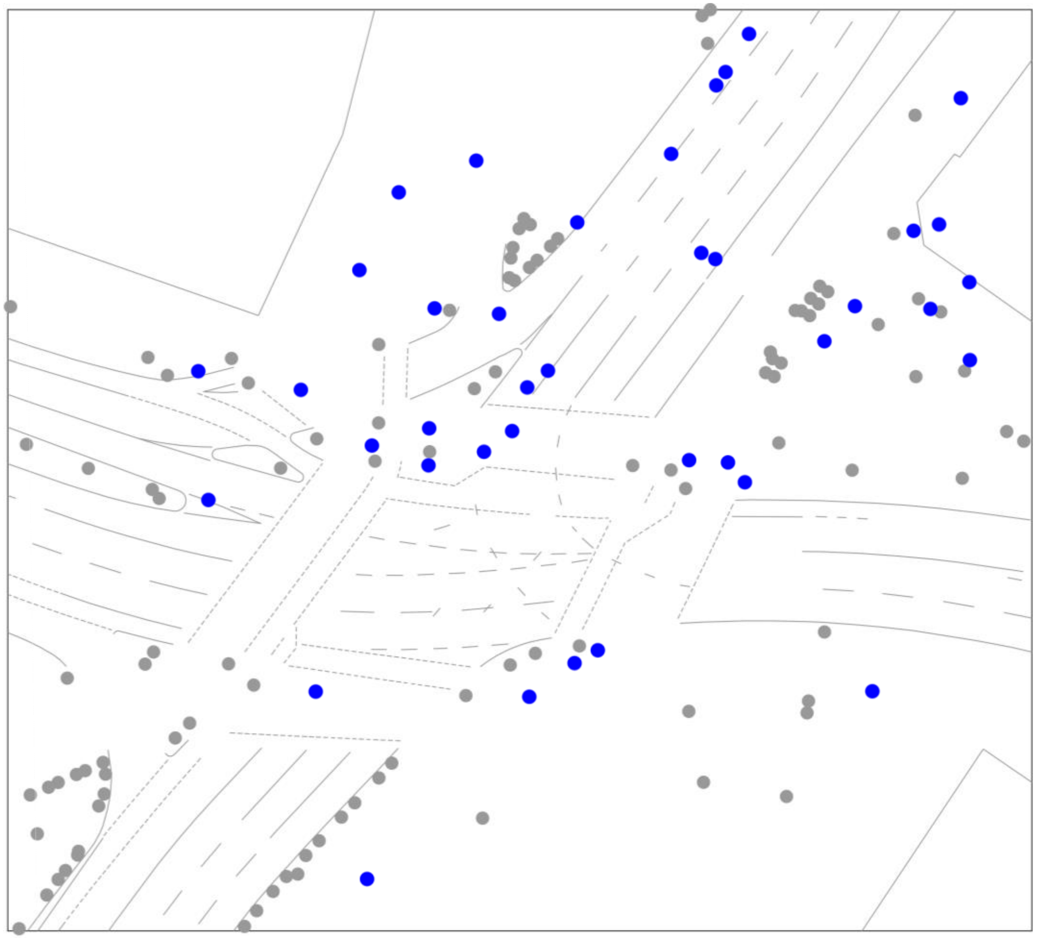


Map matching



- local map
- global map

Map matching

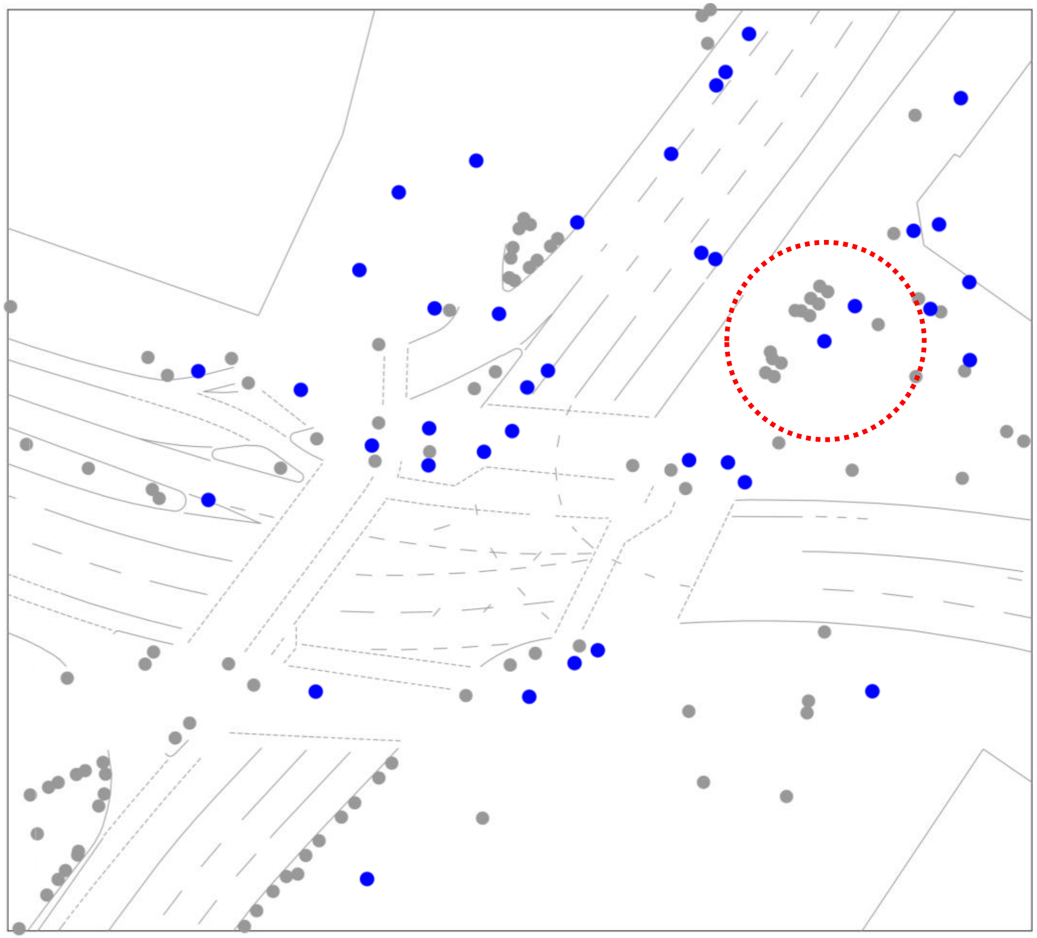


- initial projection
- generate candidate transformations
- apply
- compute cost

repeat

[Wilbers (2021)]

Map matching

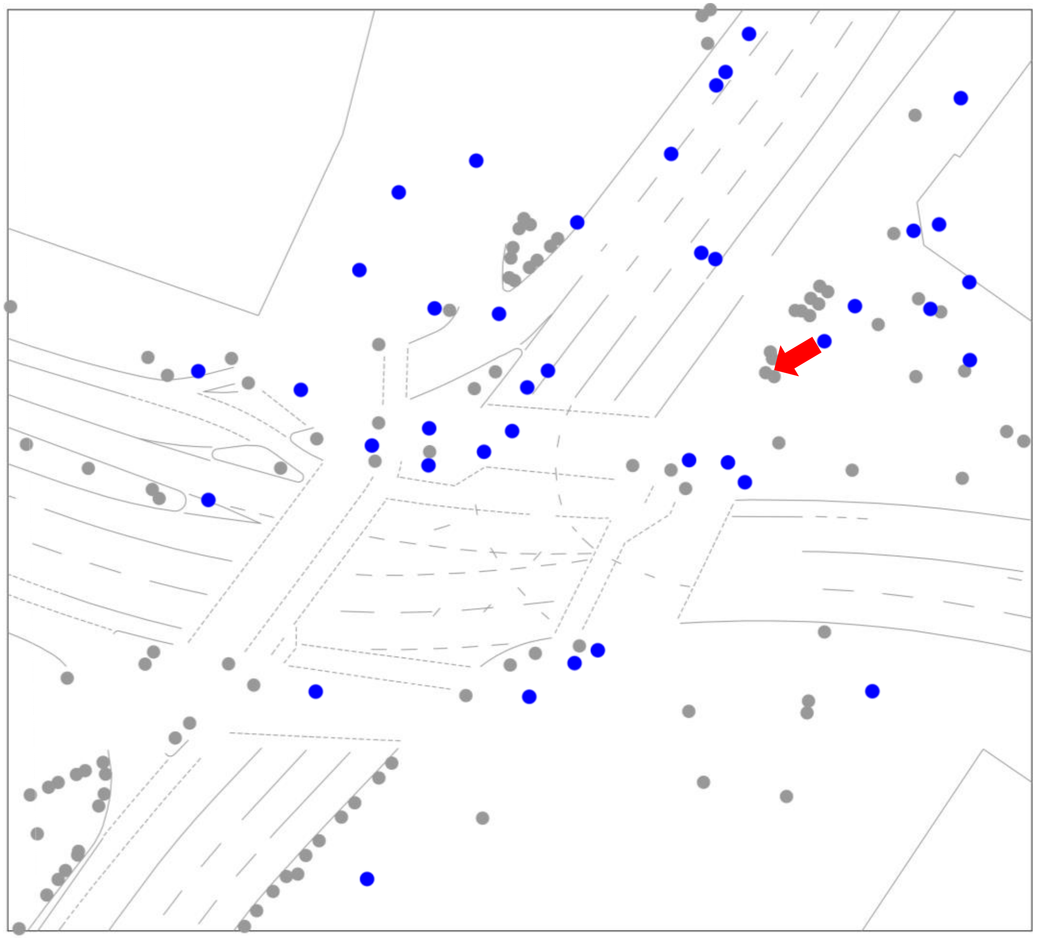


- initial projection
- generate candidate transformations
- apply
- compute cost

repeat

[Wilbers (2021)]

Map matching

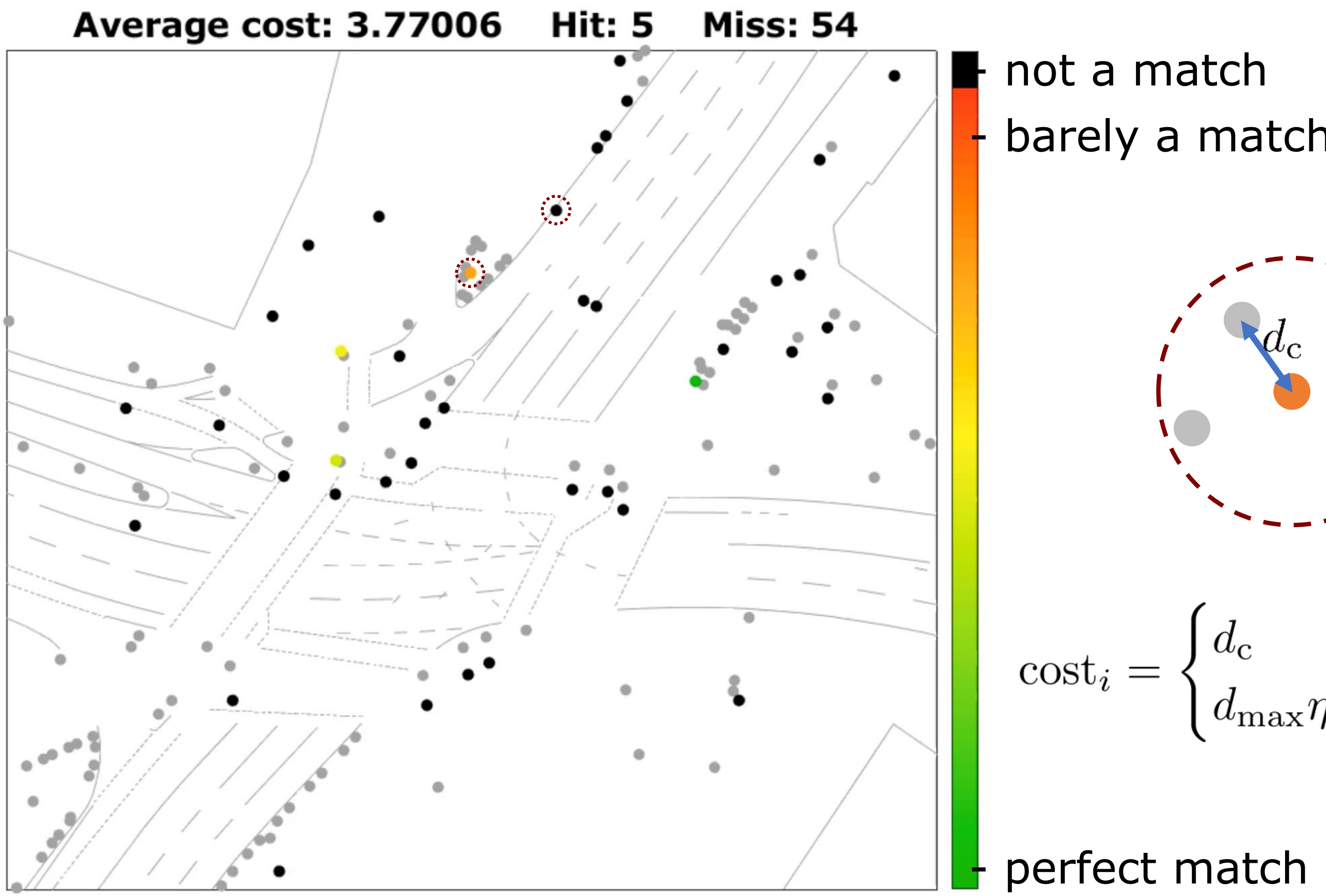


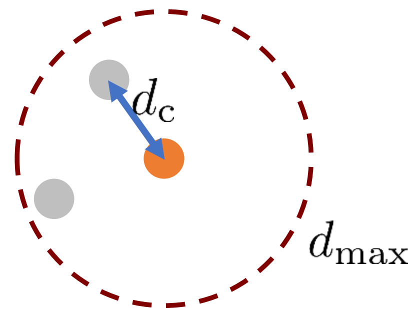
- initial projection
- generate candidate transformations
- apply
- compute cost

repeat

[Wilbers (2021)]

Map matching

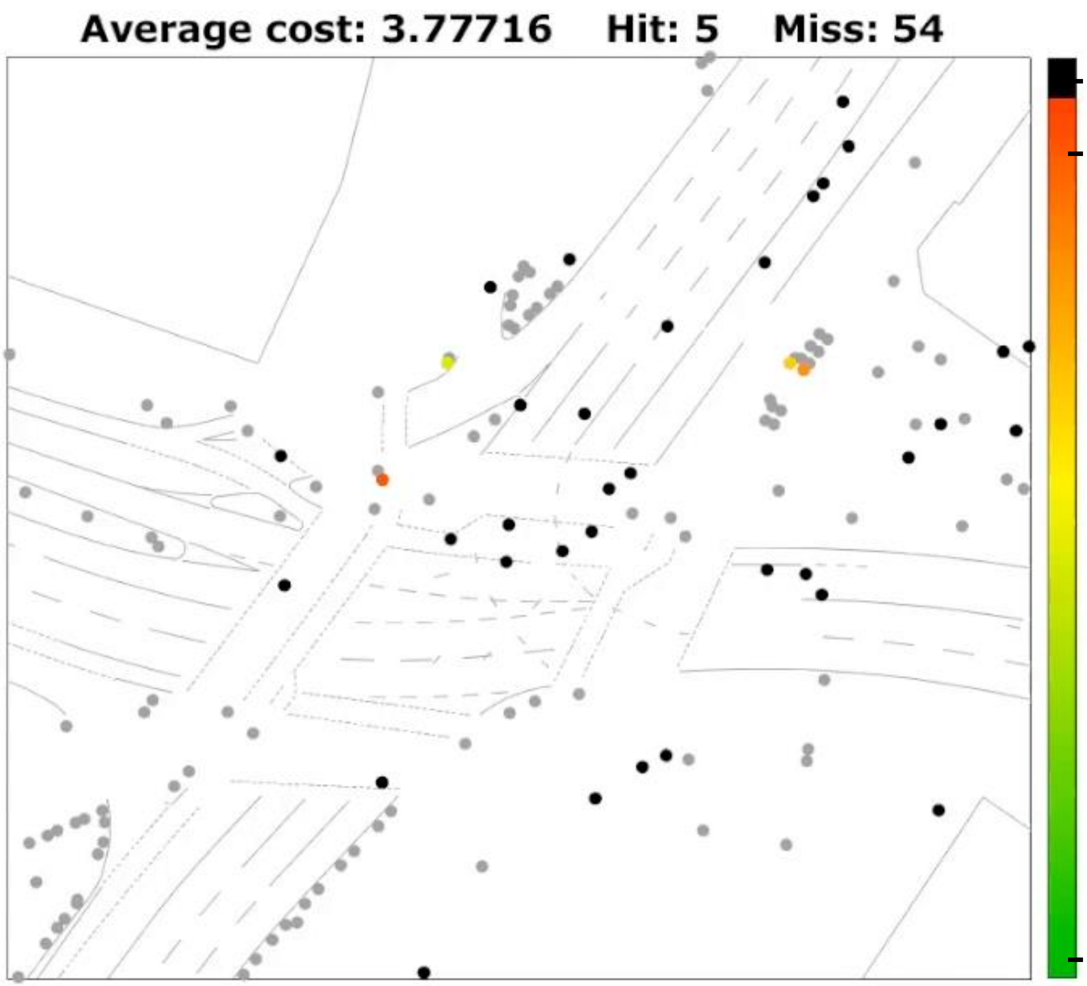


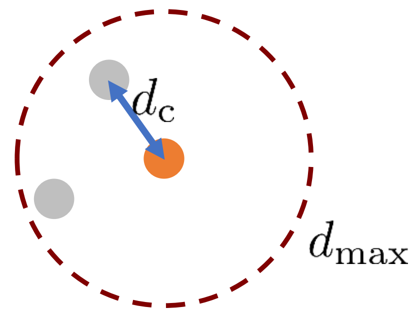


$$\mathrm{cost}_i = \begin{cases} d_{\mathrm{c}} & d_{\mathrm{c}} < d_{\max} \\ d_{\max}\eta & d_{\mathrm{c}} \geq d_{\max} \end{cases}$$

[Wilbers (2021)]

Map matching

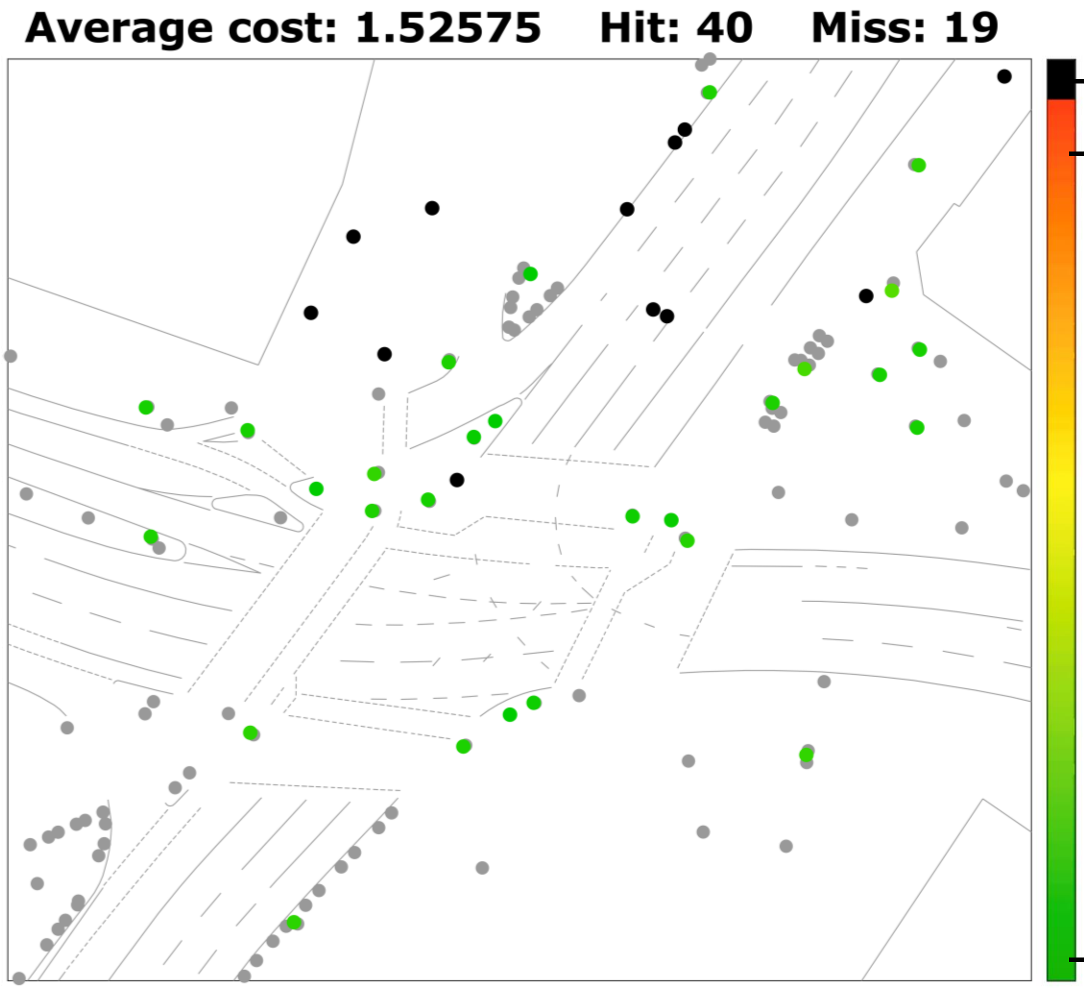


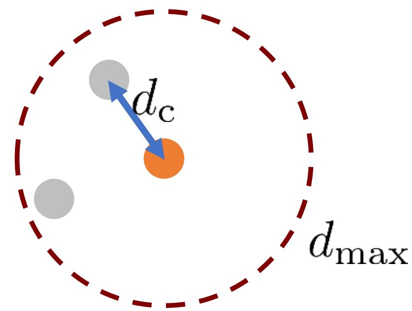


$$\mathrm{cost}_i = \begin{cases} d_{\mathrm{c}} & d_{\mathrm{c}} < d_{\max} \\ d_{\max}\eta & d_{\mathrm{c}} \geq d_{\max} \end{cases}$$

[Wilbers (2021)]

Map matching

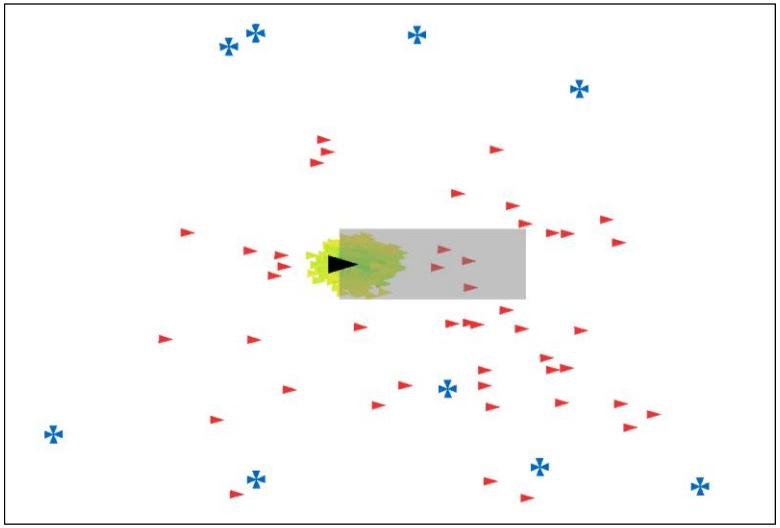




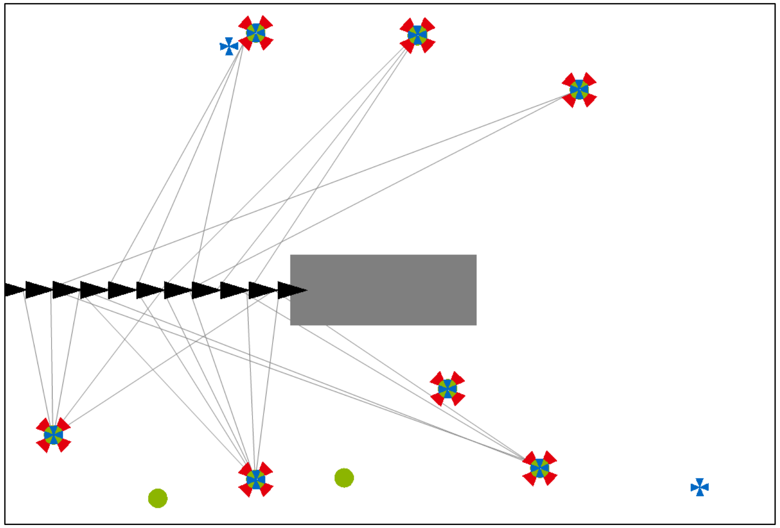
$$\mathrm{cost}_i = \begin{cases} d_\mathrm{c} & d_\mathrm{c} < d_\mathrm{max} \\ d_\mathrm{max}\eta & d_\mathrm{c} \geq d_\mathrm{max} \end{cases}$$

[Wilbers (2021)]

State estimation



Particle Filter



Graph-based approaches

- other: EKF, Bayesian methods, Deep-learning techniques, …

State estimation

$$\boldsymbol{x}_\mathcal{T}^* = \underset{\boldsymbol{x}_\mathcal{T}}{\operatorname{argmax}}\, p(\underline{\boldsymbol{x}_\mathcal{T}} \mid \underline{\boldsymbol{z}^{\mathrm{lm}}}, \underline{\boldsymbol{z}^{\mathrm{odo}}}, \underline{\boldsymbol{z}^{\mathrm{abs}}}, \underline{\boldsymbol{m}})$$

- map
- GNSS
- odometry
- landmark measurements
- trajectory + landmarks

$$\boldsymbol{x}_\mathcal{T} = [\boldsymbol{x}_1^p, \ldots, \boldsymbol{x}_N^p, \boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_M^l]^\top$$

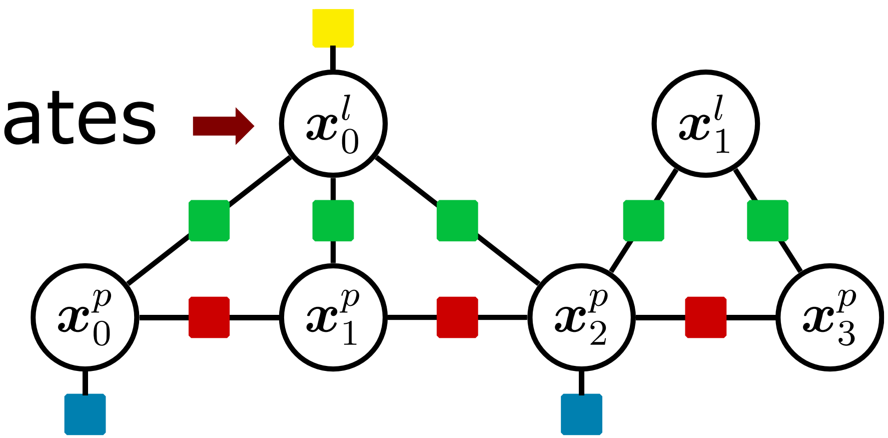
$$\boldsymbol{x}_\mathcal{T}^* = \underset{\boldsymbol{x}_\mathcal{T}}{\operatorname{argmin}} \sum_i \rho\left(\boldsymbol{e}_k(\boldsymbol{x}_\mathcal{T}, \boldsymbol{z}_i^k)^\top \Omega_i \boldsymbol{e}_k(\boldsymbol{x}_\mathcal{T}, \boldsymbol{z}_i^k)\right)$$

→ solve with, e.g., Gauss-Newton

Factor graph optimization

$$\boldsymbol{x}_\tau^* = \underset{\boldsymbol{x}_\tau}{\operatorname{argmin}} \sum_i \rho\left(\boldsymbol{e}_k(\boldsymbol{x}_\tau, \boldsymbol{z}_i^k)^\top \Omega_i \boldsymbol{e}_k(\boldsymbol{x}_\tau, \boldsymbol{z}_i^k)\right)$$

states

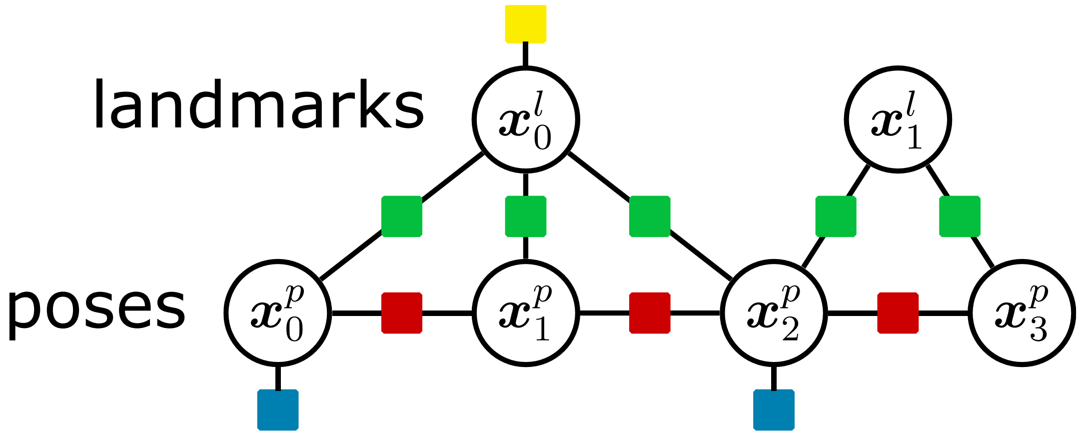


factors $\{\boldsymbol{z}_i^k, \Omega_i, \mathrm{e}_k(\boldsymbol{x}_\tau, \boldsymbol{z}_i^k)\}$

- error function
- information matrix
- measurement

Factor graph optimization

$$\boldsymbol{x}_{\mathcal{T}}^{*} = \underset{\boldsymbol{x}_{\mathcal{T}}}{\operatorname{argmin}} \sum_{i} \rho\left(\boldsymbol{e}_k(\boldsymbol{x}_{\mathcal{T}}, \boldsymbol{z}_i^k)^\top \Omega_i \boldsymbol{e}_k(\boldsymbol{x}_{\mathcal{T}}, \boldsymbol{z}_i^k)\right)$$

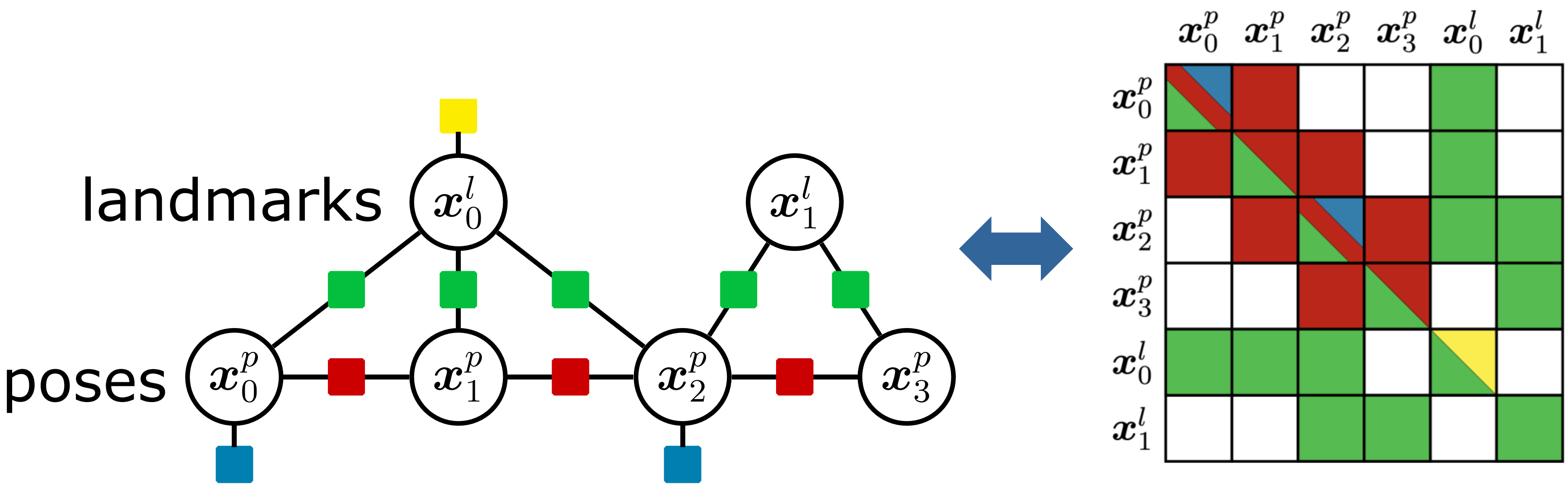


factors: GNSS, map, odometry, landmark detections

Factor graph optimization

$$\boldsymbol{x}_{\mathcal{T}}^{*} = \underset{\boldsymbol{x}_{\mathcal{T}}}{\operatorname{argmin}} \sum_{i} \rho\left(\boldsymbol{e}_k(\boldsymbol{x}_{\mathcal{T}}, \boldsymbol{z}_i^k)^{\top} \Omega_i \boldsymbol{e}_k(\boldsymbol{x}_{\mathcal{T}}, \boldsymbol{z}_i^k)\right)$$

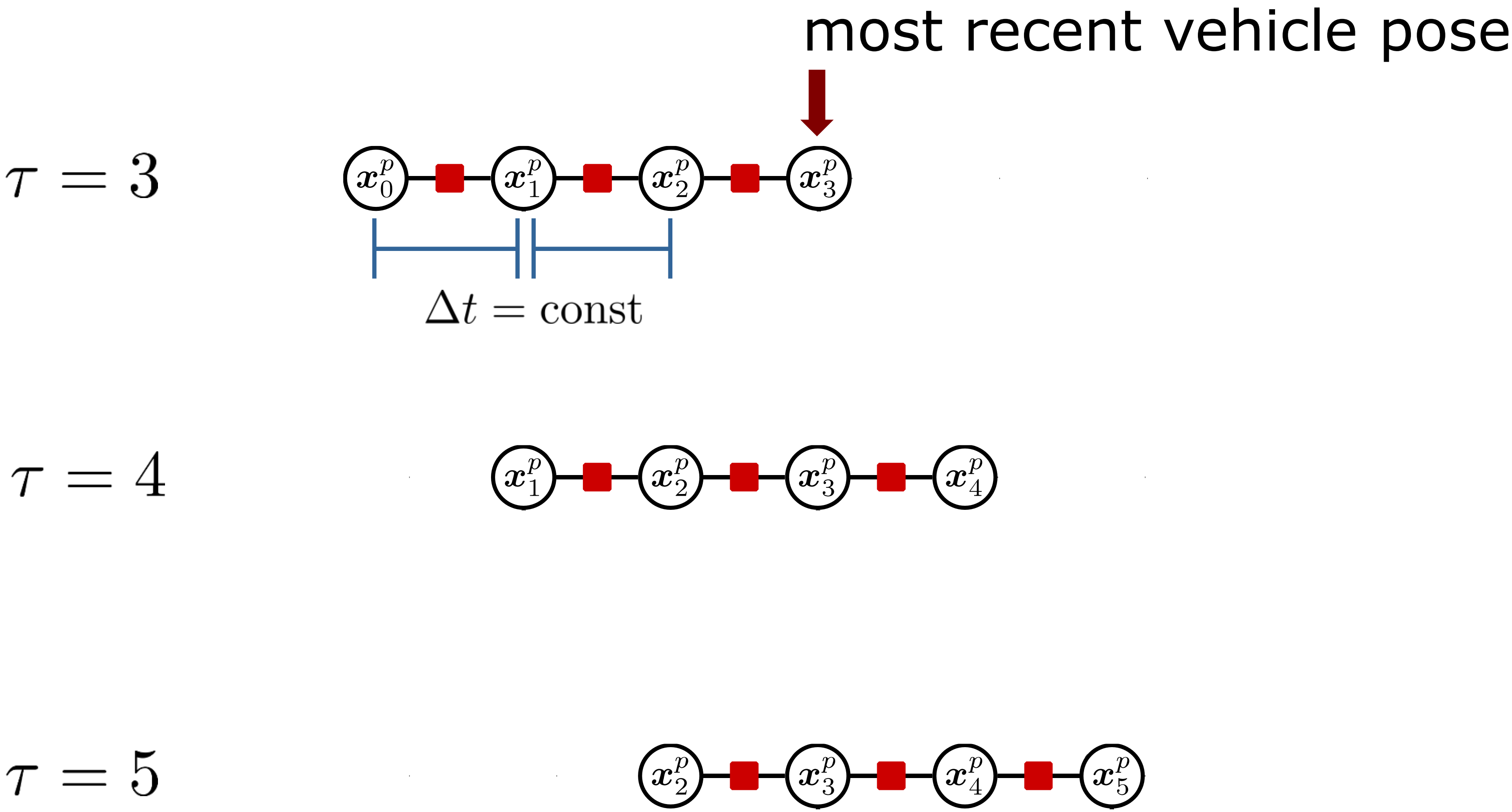
Gauss-Newton update: $\boldsymbol{x}_{\mathcal{T}}^{*} = \breve{\boldsymbol{x}}_{\mathcal{T}} + \Delta \boldsymbol{x}_{\mathcal{T}}^{*}$



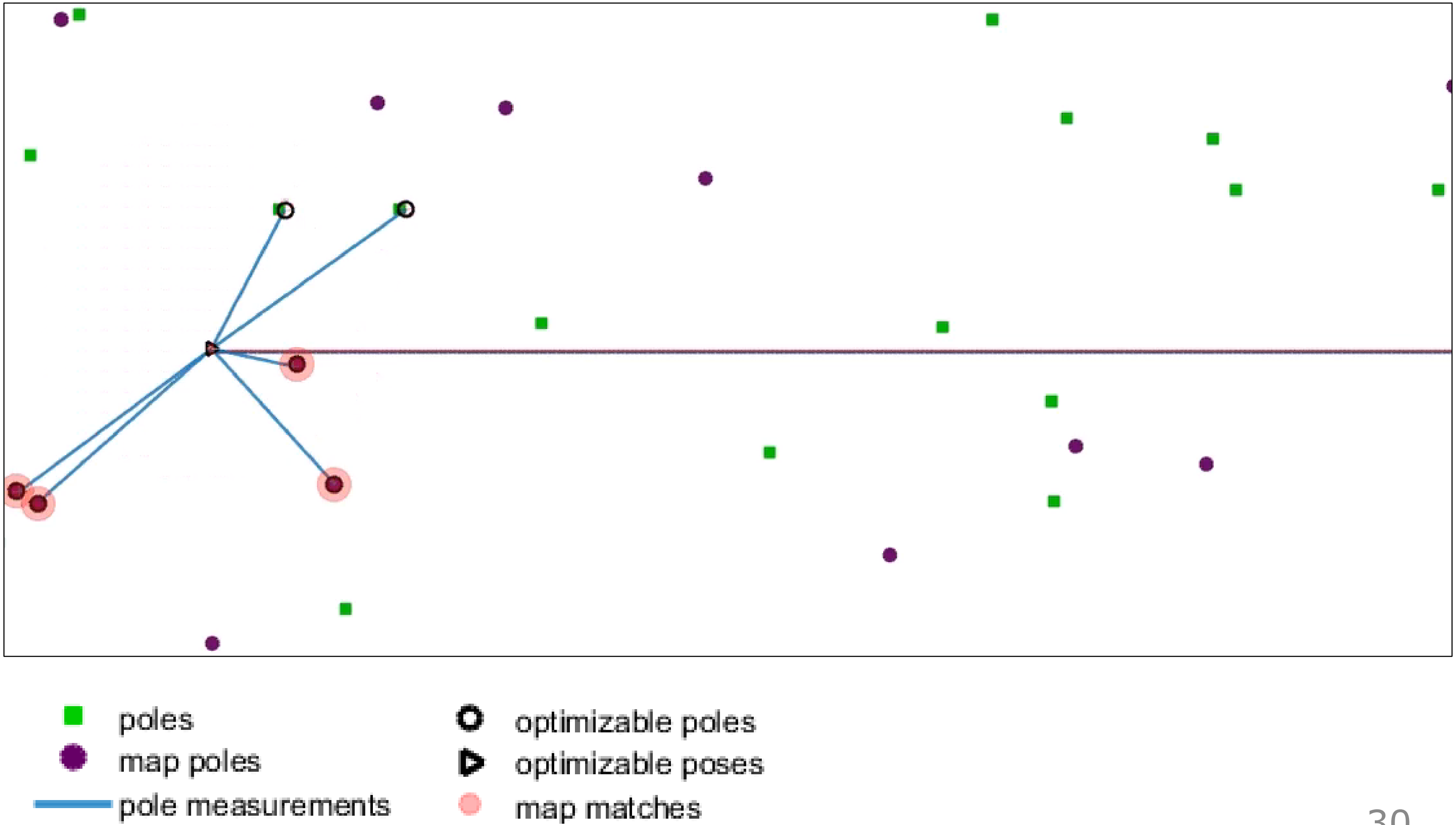
system matrix H in

$$H \Delta \boldsymbol{x}_{\mathcal{T}}^{*} = -\boldsymbol{b}$$

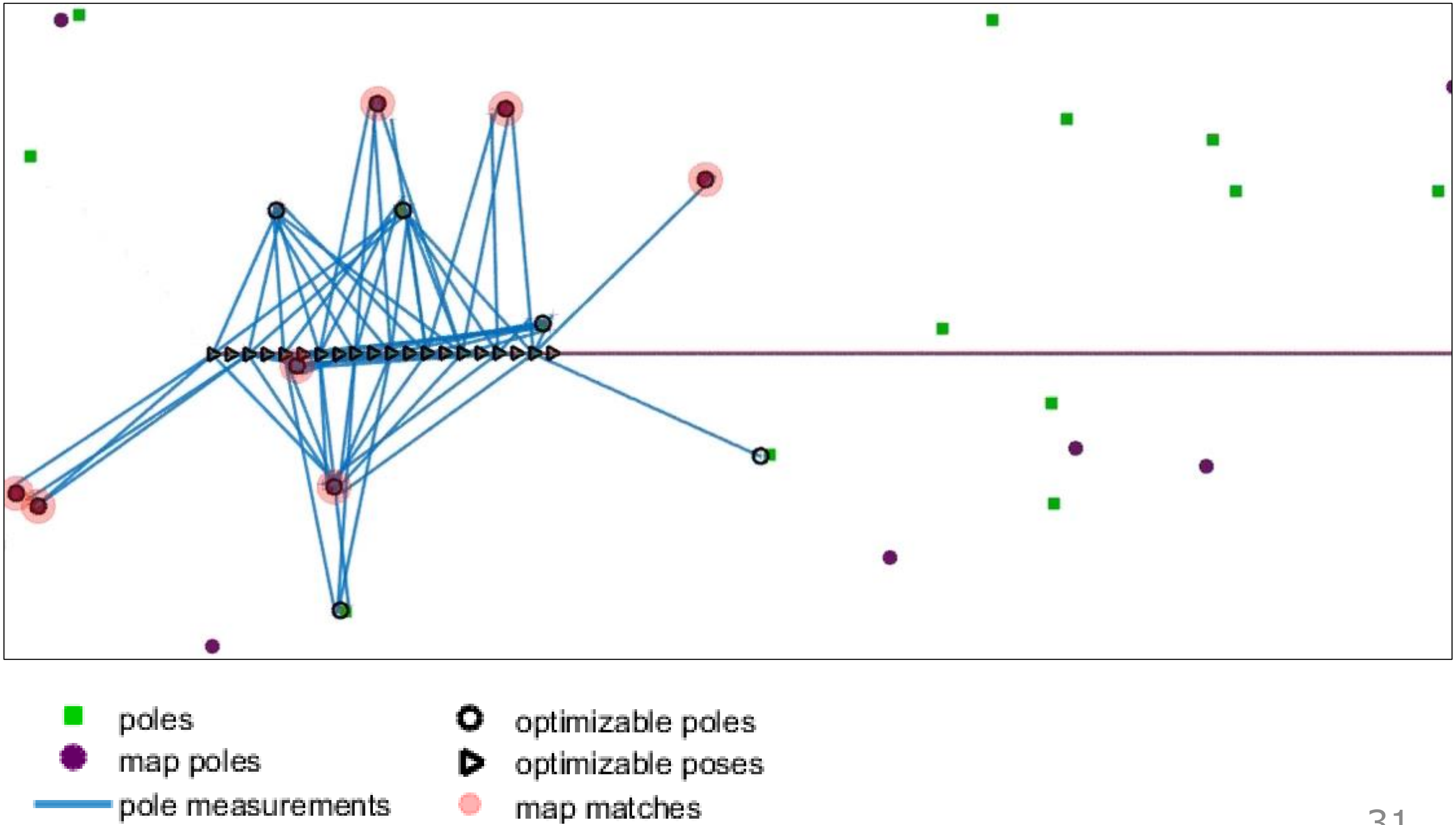
Sliding window factor graphs



Toy example

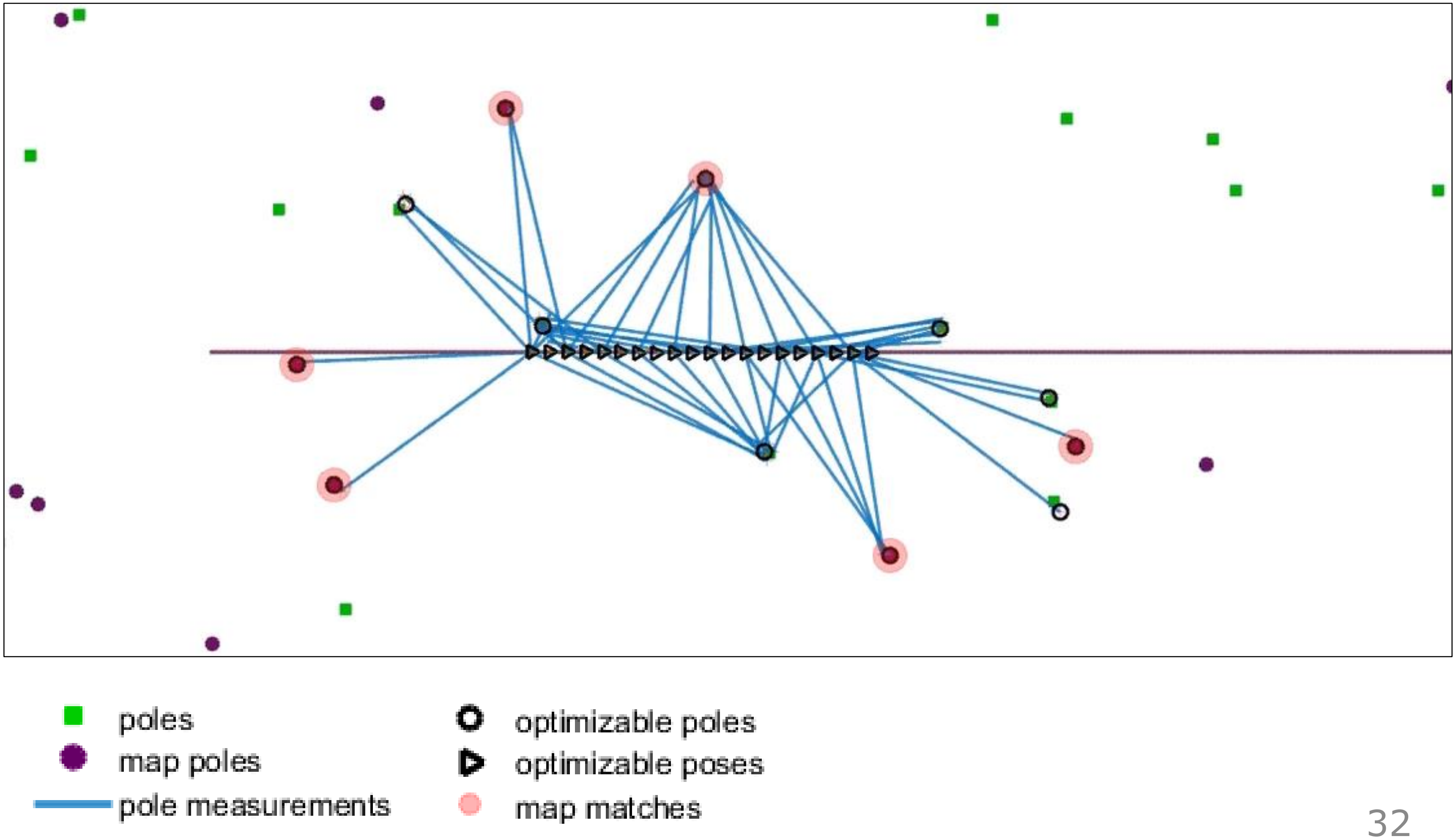

poles
map poles
pole measurements
optimizable poles
optimizable poses
map matches

Toy example

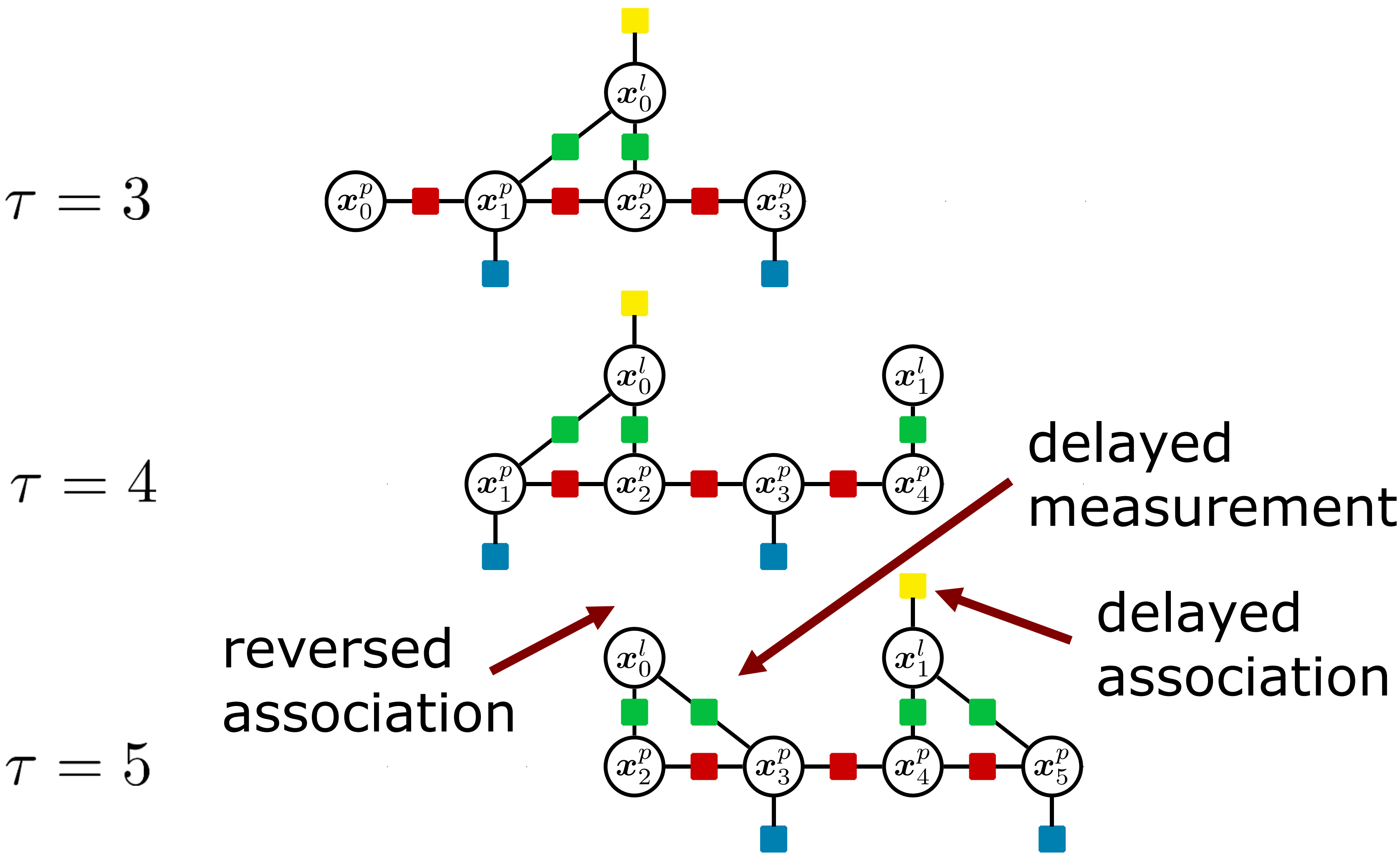


poles
map poles
pole measurements
optimizable poles
optimizable poses
map matches

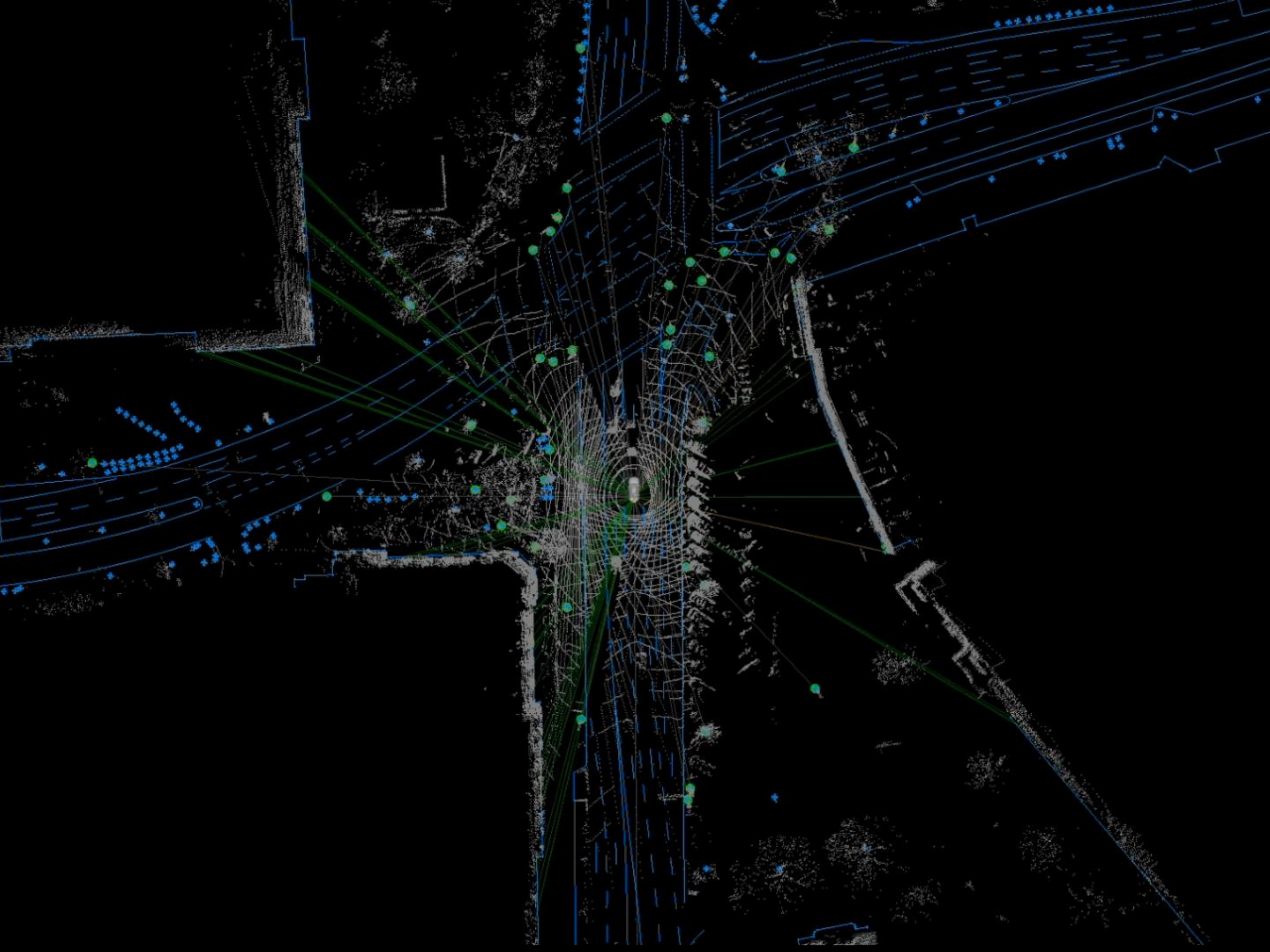
Toy example

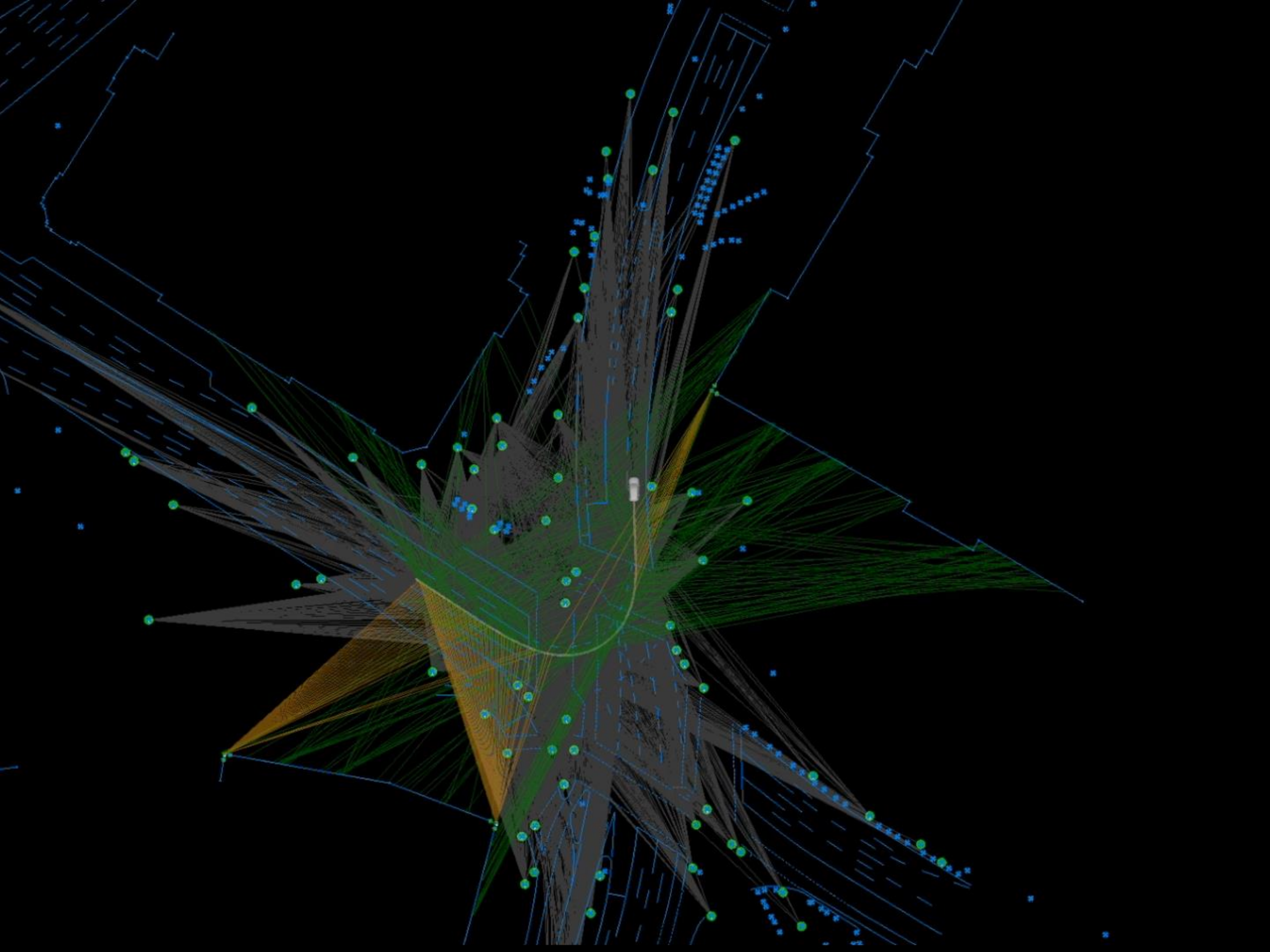

poles
map poles
pole measurements
optimizable poles
optimizable poses
map matches

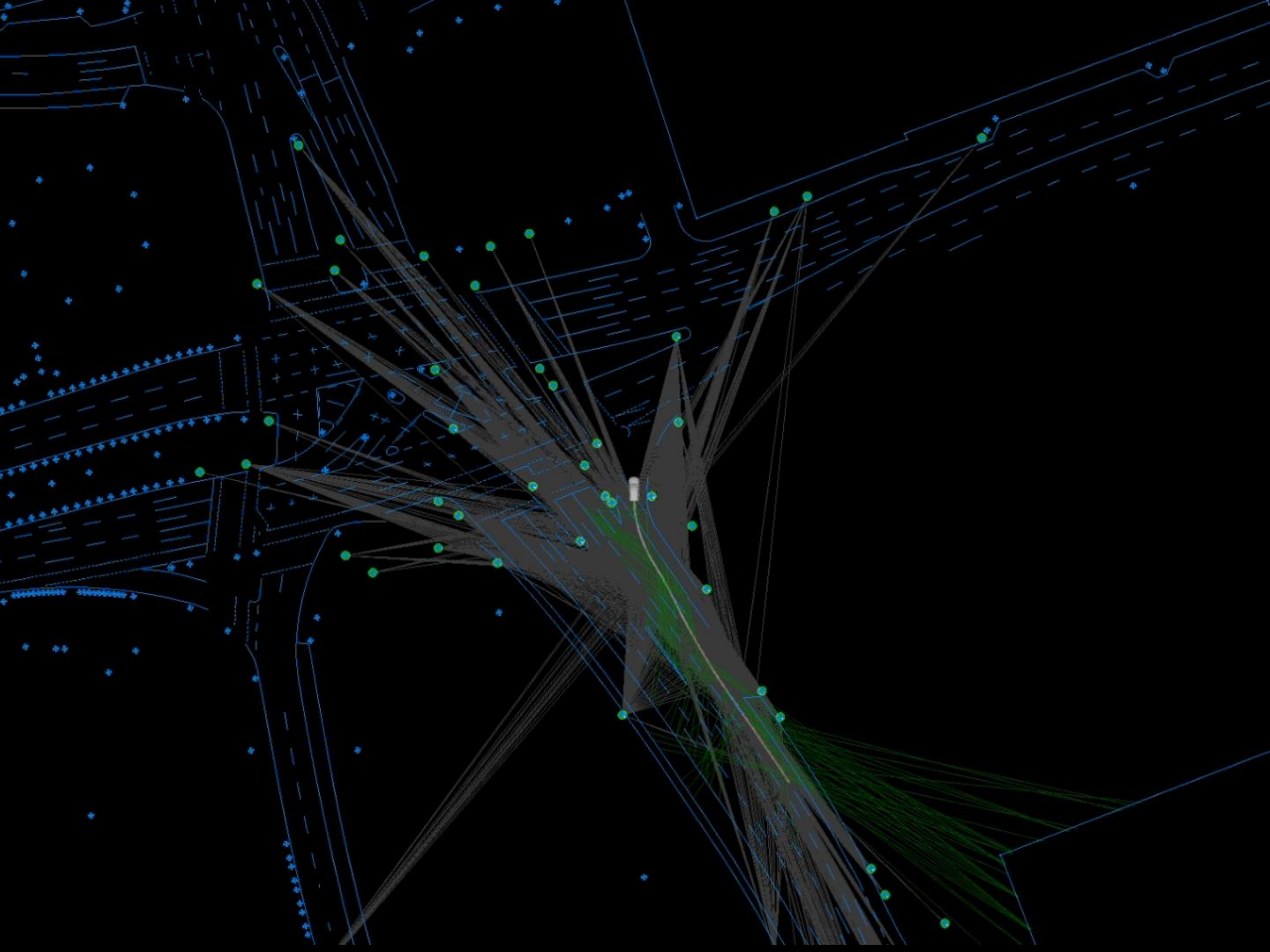
Sliding window factor graphs



[see Wilbers (2021)]





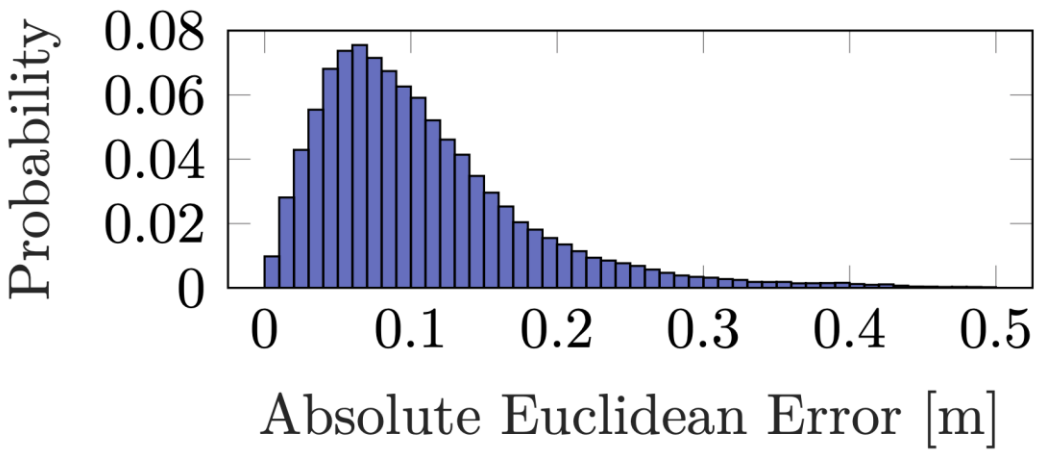


Dataset (255 km)

[Images courtesy of Volkswagen AG]

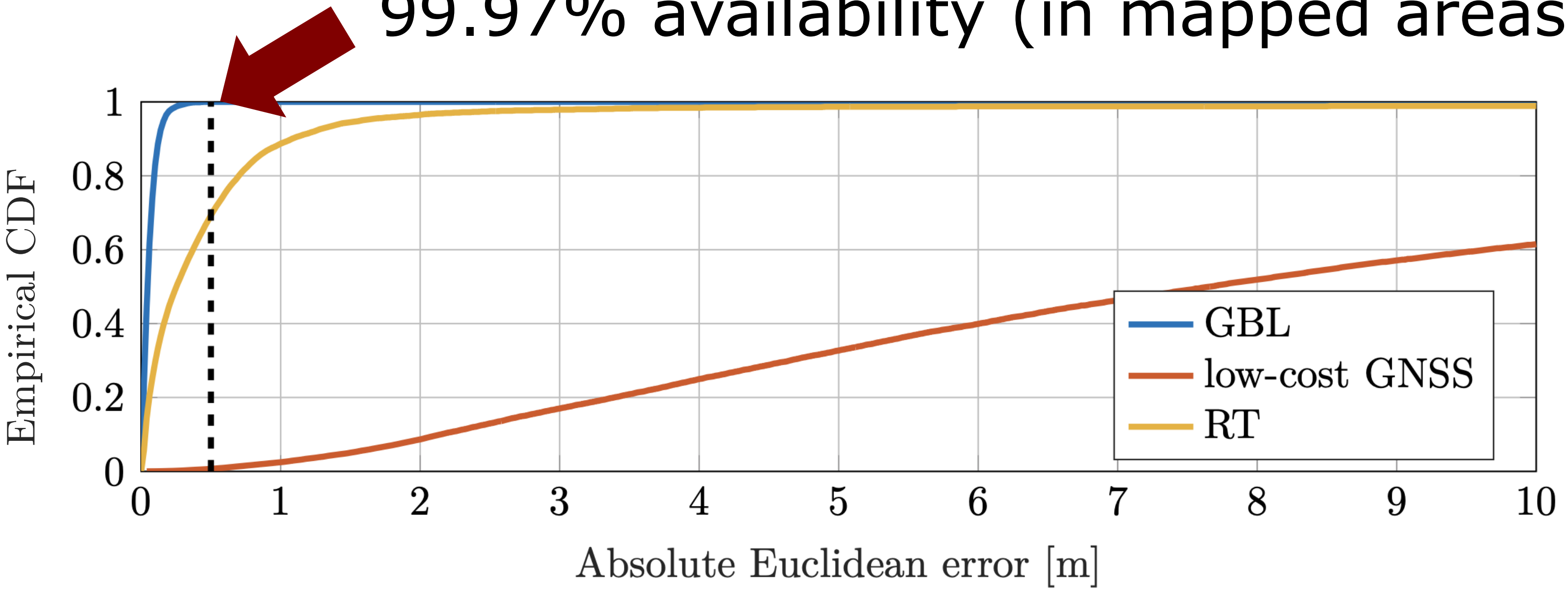
Localization results

error type	GBL	low-cost GNSS	RT3000
lateral	**0.06** m	7.08 m	0.43 m
longitudinal	**0.08** m	5.56 m	0.39 m
heading	**0.11** deg	31.36 deg	1.05 deg
Euclidean	**0.11** m	10.12 m	0.65 m



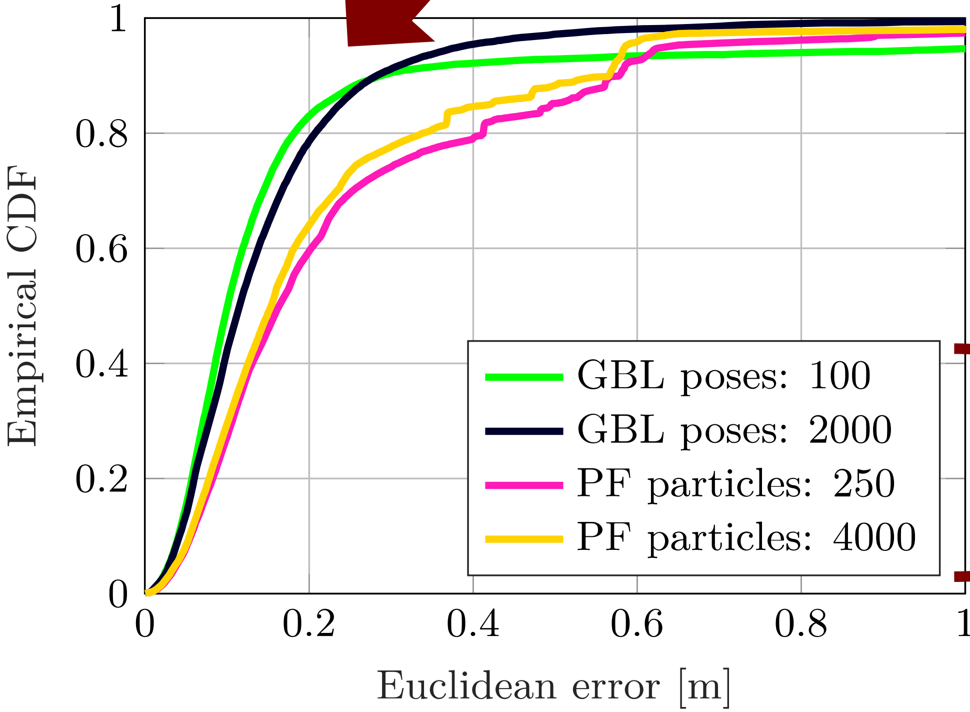
Localization results

99.97% availability (in mapped areas)



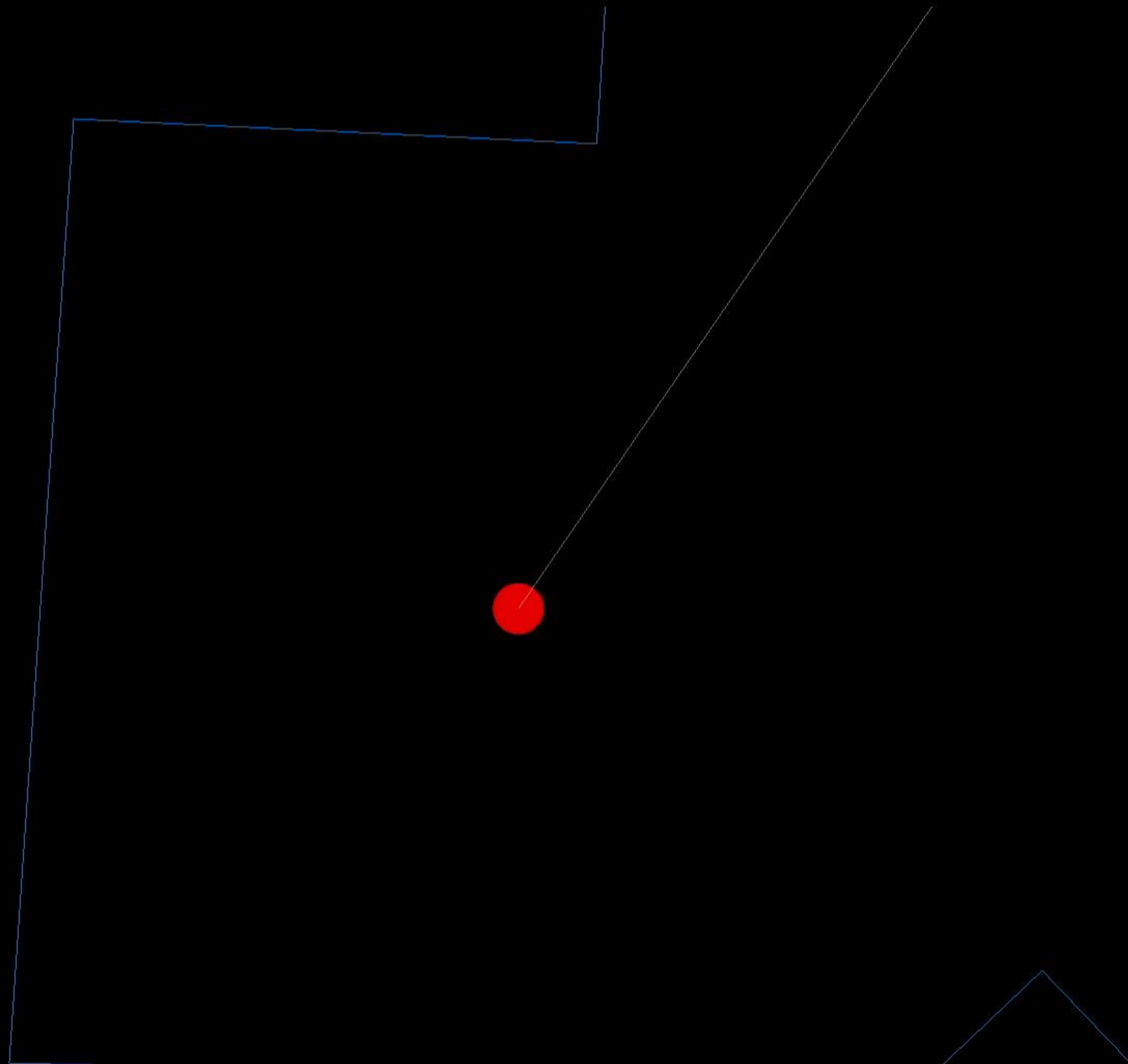
Comparison to a particle filter

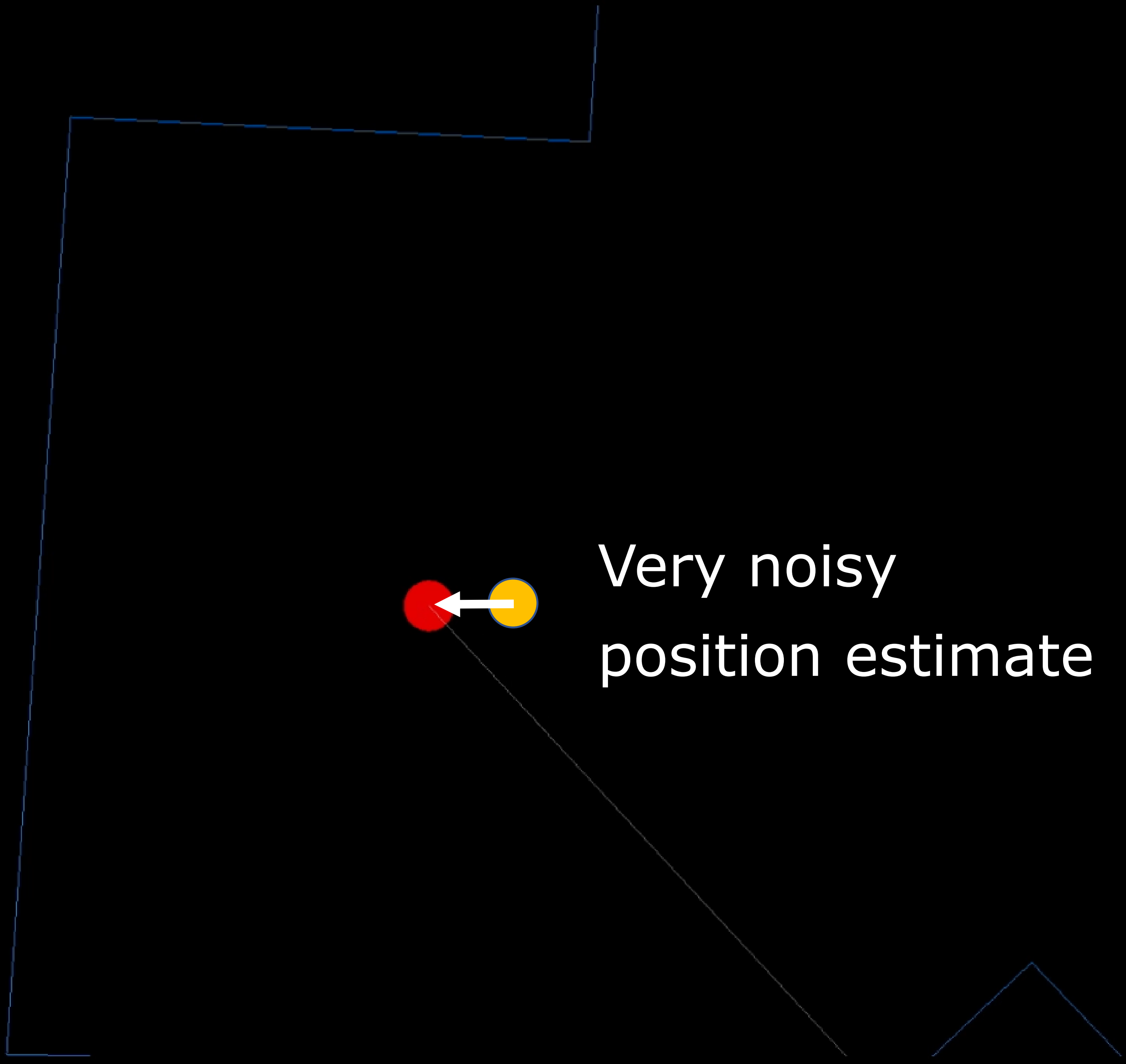
GBL (ours) outperforms PF
in all tested cases



different state
complexities

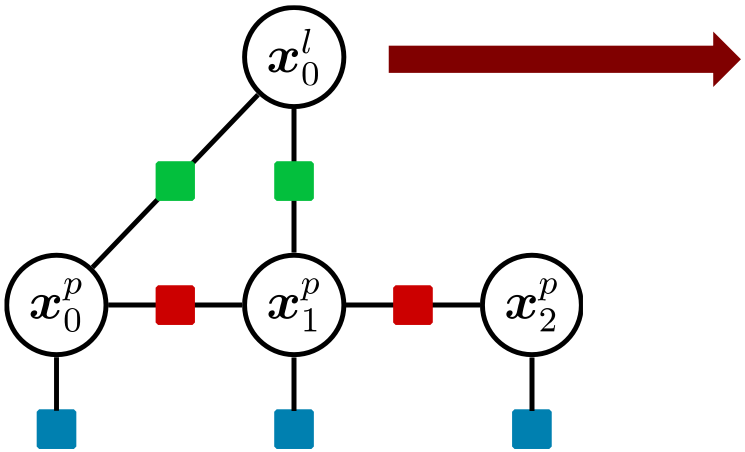
based on 16km dataset




Very noisy
position estimate

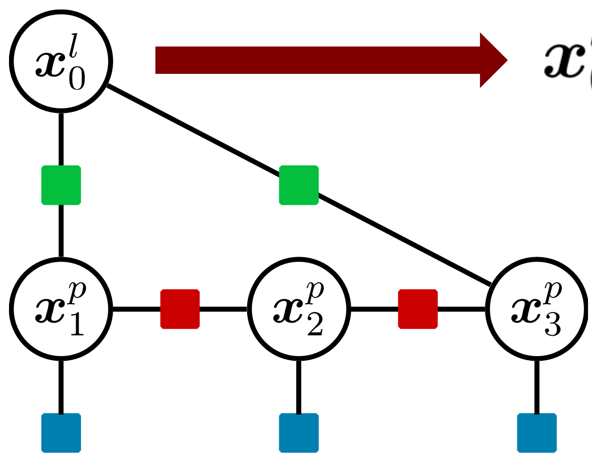
Estimating landmarks

$\tau = 2$



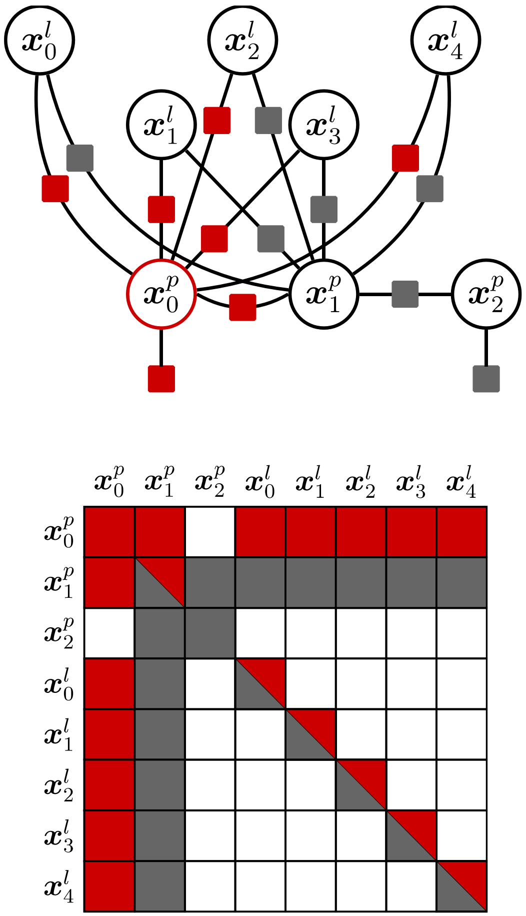
$$\boldsymbol{x}_0^l = \begin{bmatrix} 1.1 \\ 0 \end{bmatrix}$$

$\tau = 3$

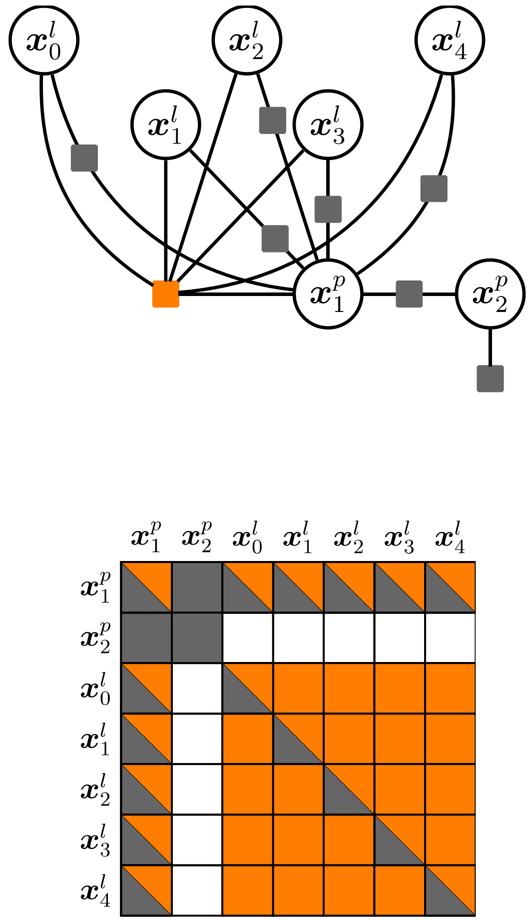


$$\boldsymbol{x}_0^l = \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}$$

Marginalization (worst-case)

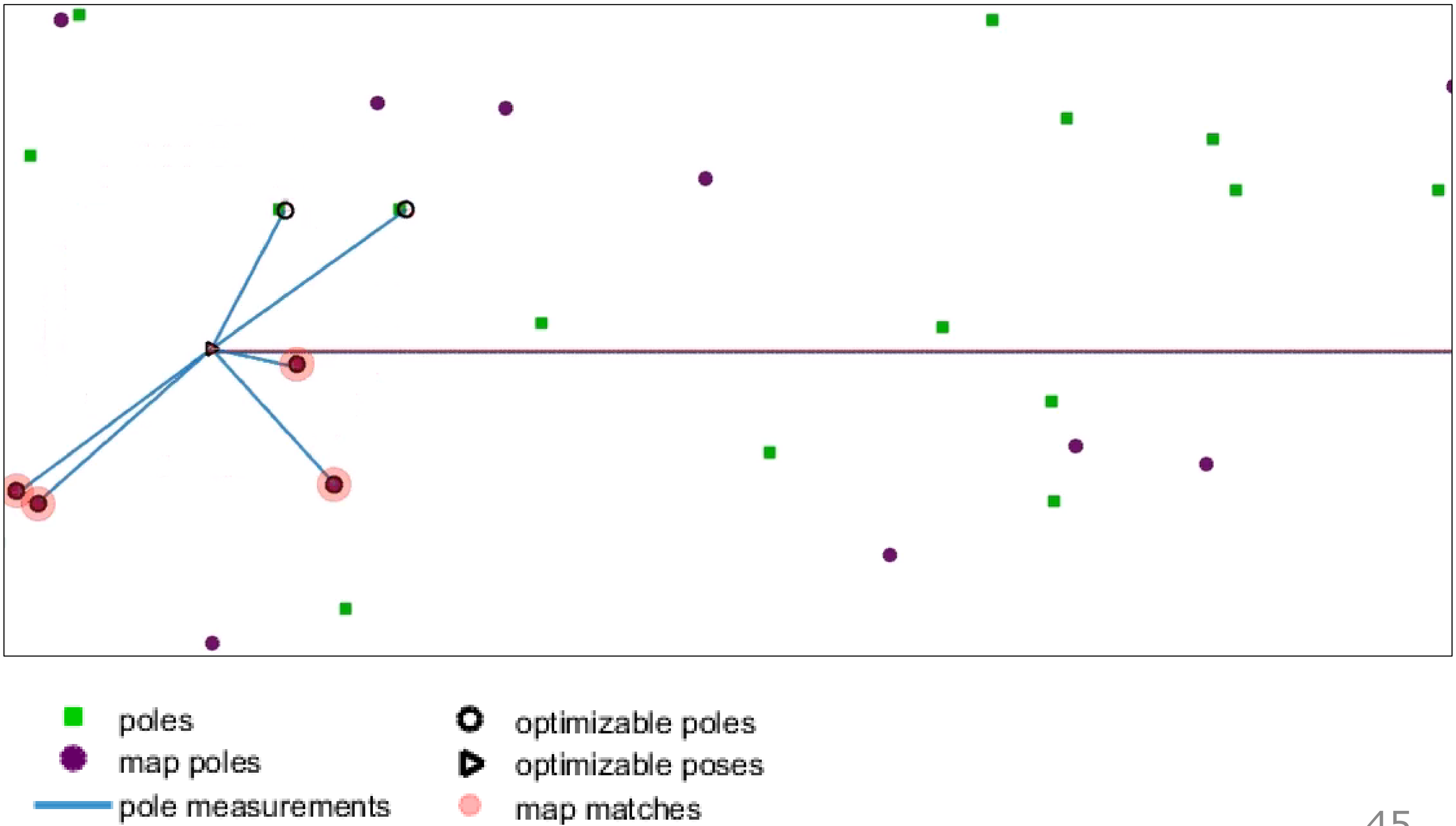


before marginalization

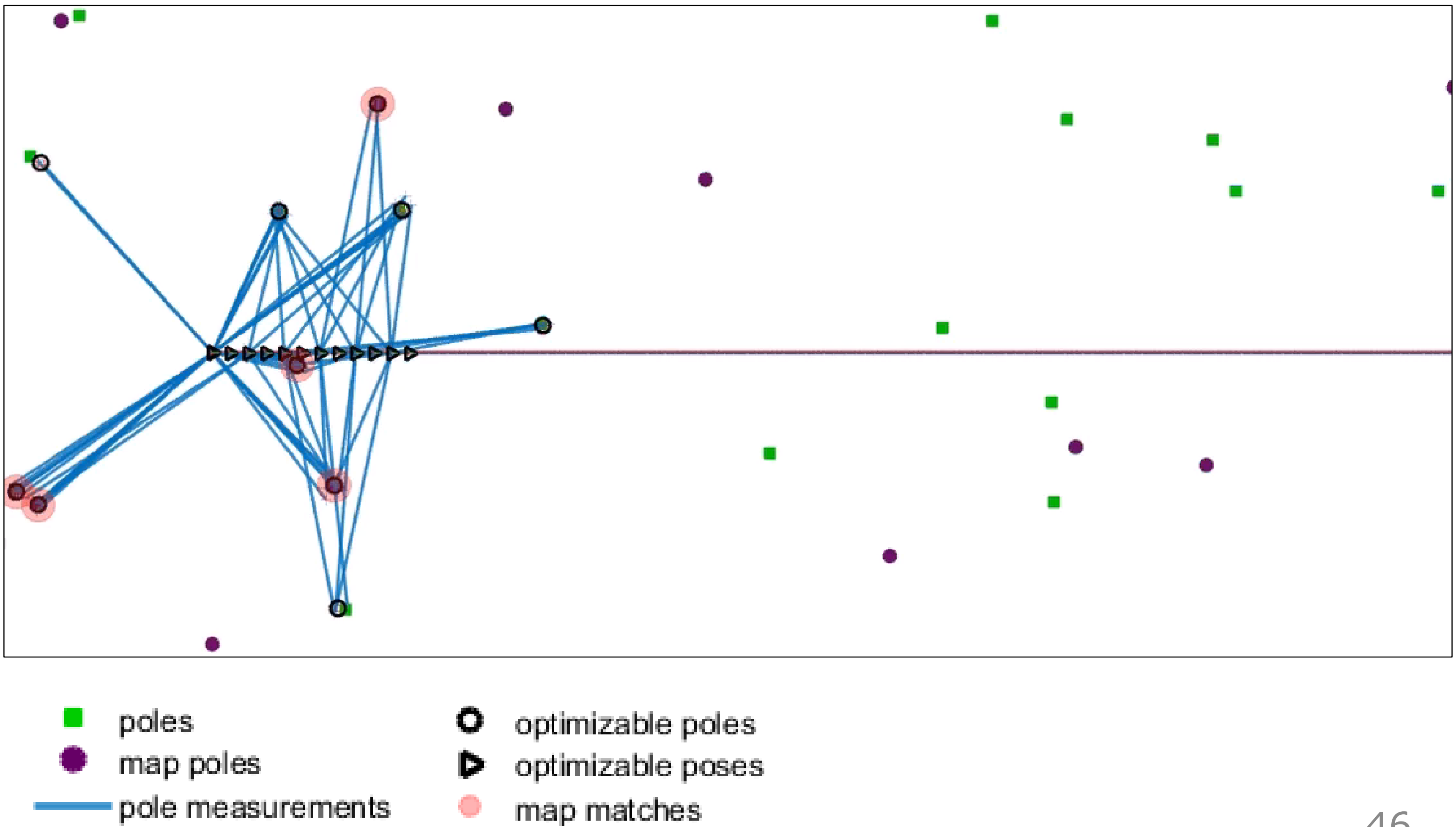


after marginalization

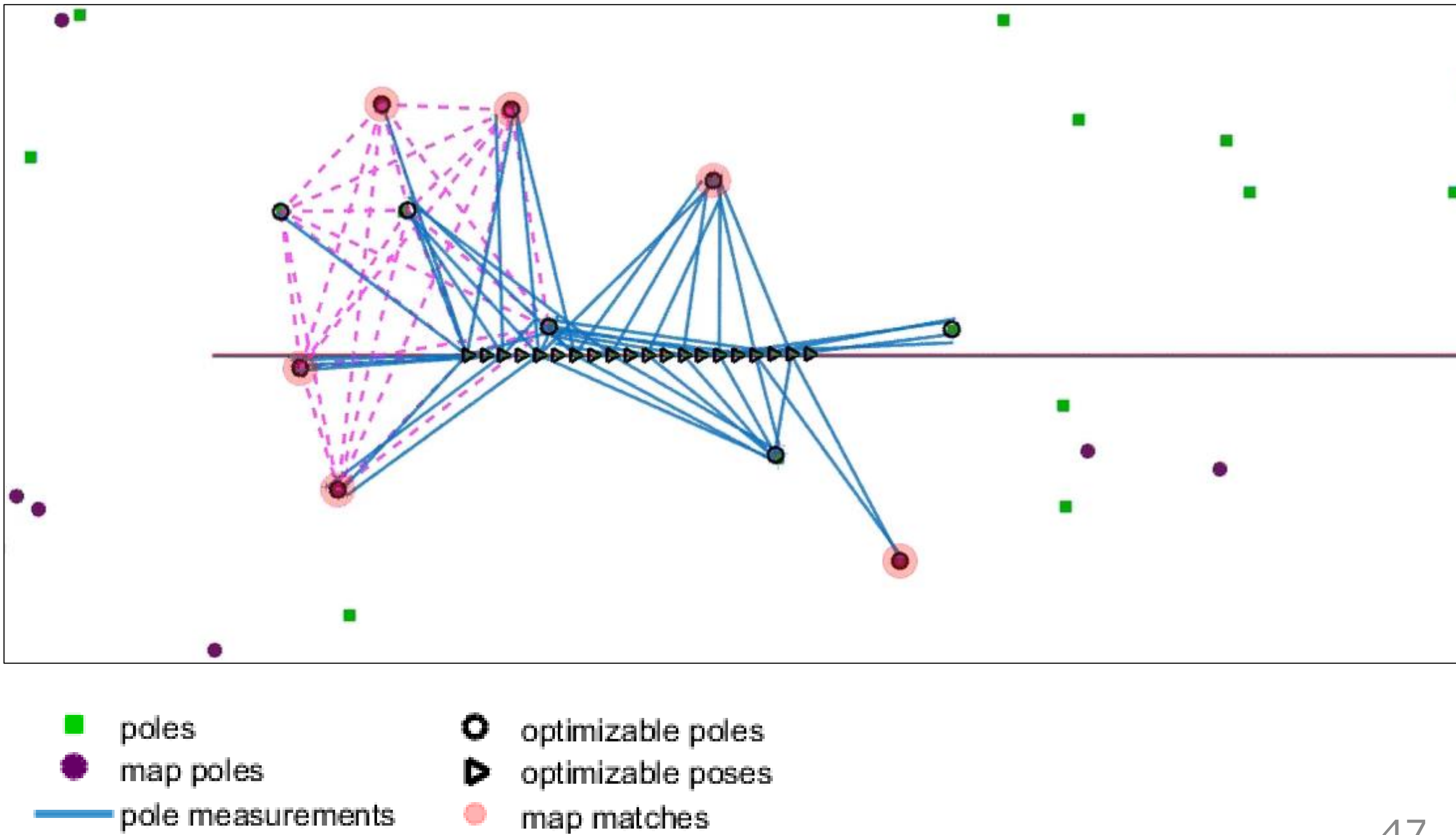
Dense Marginalization - Toy Example



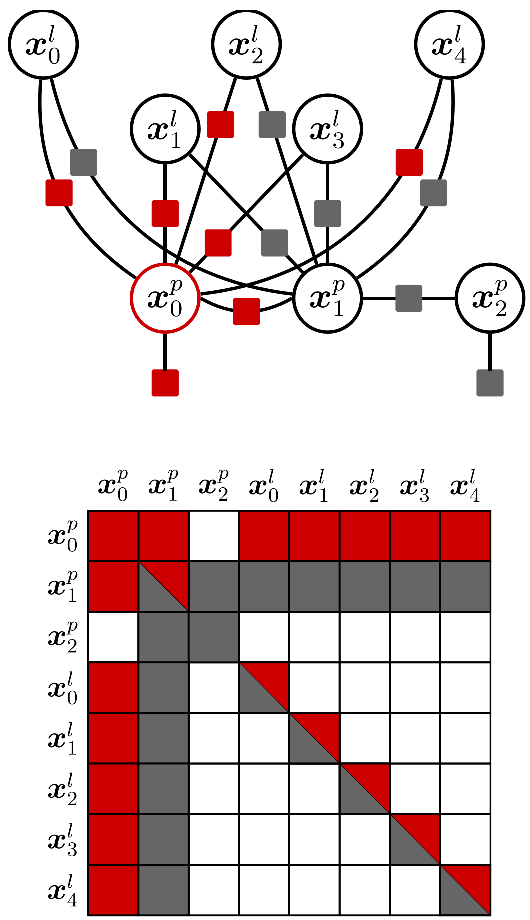
Dense Marginalization - Toy Example



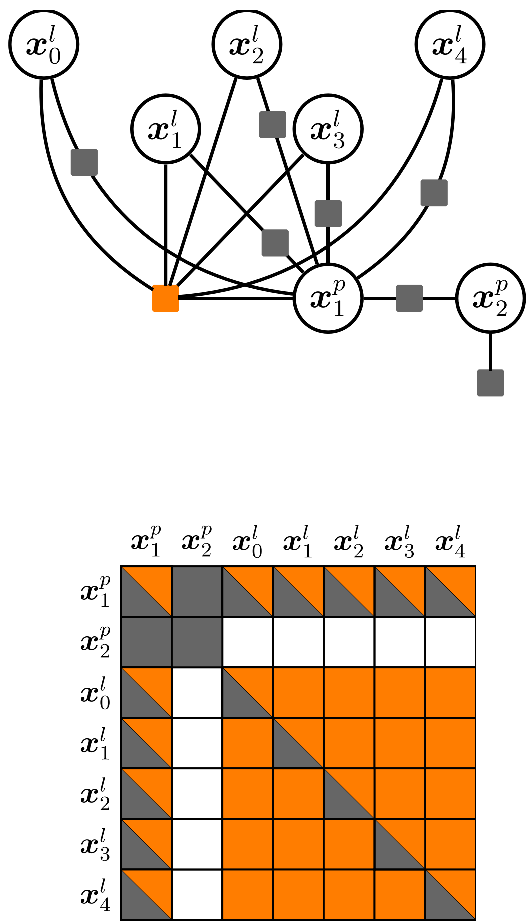
Dense Marginalization - Toy Example



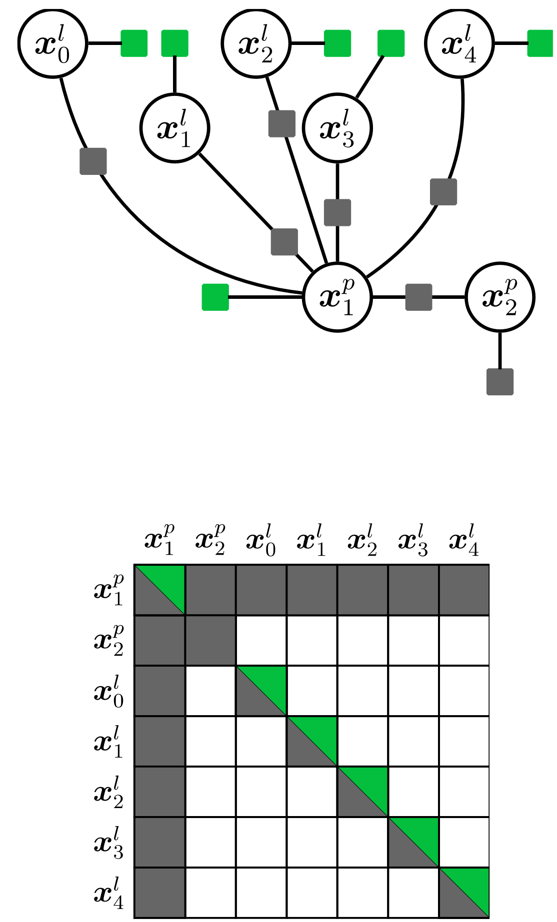
Sparse global priors



before marginalization



after marginalization



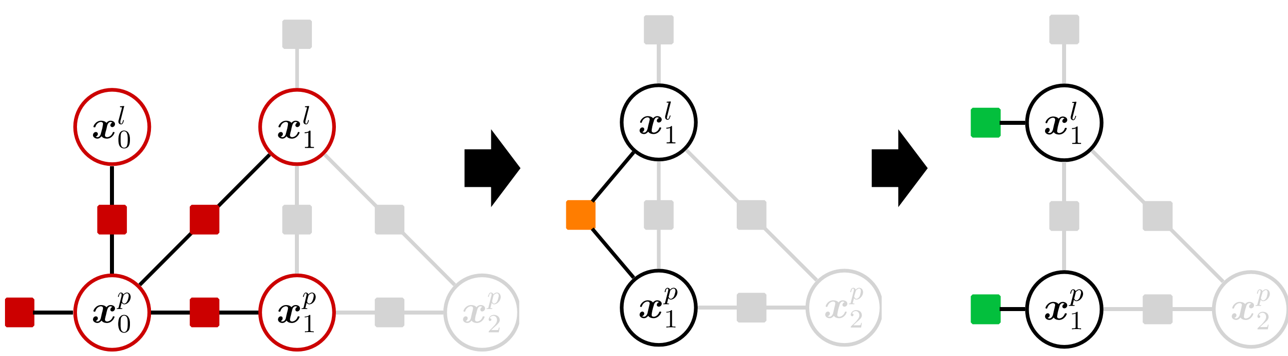
Wilbers et al. (2019)

Sparse global priors

Sparsifying the marginalization prior:

$$\underset{\mu_a,\Omega_a}{\operatorname{argmin}}\ \mathcal{D}_{KL}\left(\mathcal{N}\left(\breve{\boldsymbol{x}}_n, H_t^{-1}\right) \,\|\, \prod_i \underbrace{\mathcal{N}\left(\mu_{a_i}, \Omega_{a_i}^{-1}\right)}_{\text{individual priors}}\right)$$

➡ closed form solution: $\mu_a^* = \breve{\boldsymbol{x}}_n \quad \Omega_{a_i}^* = \{H_t^{-1}\}_i^{-1}$

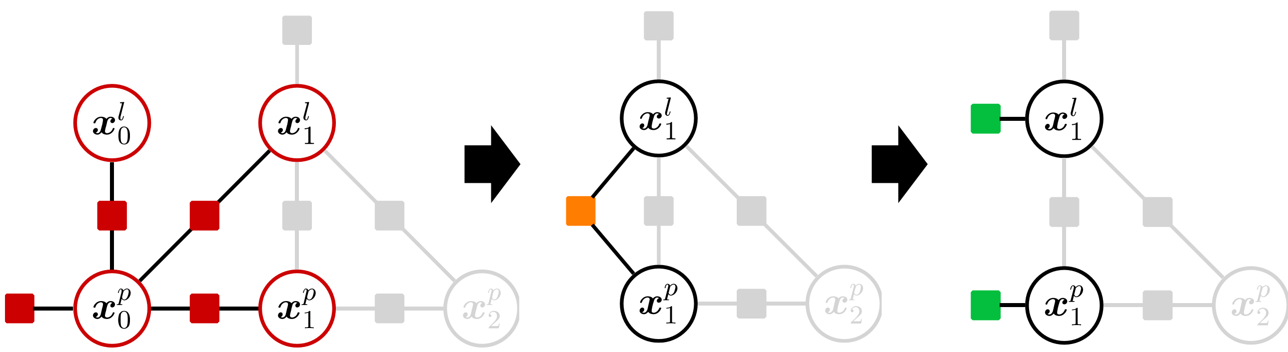


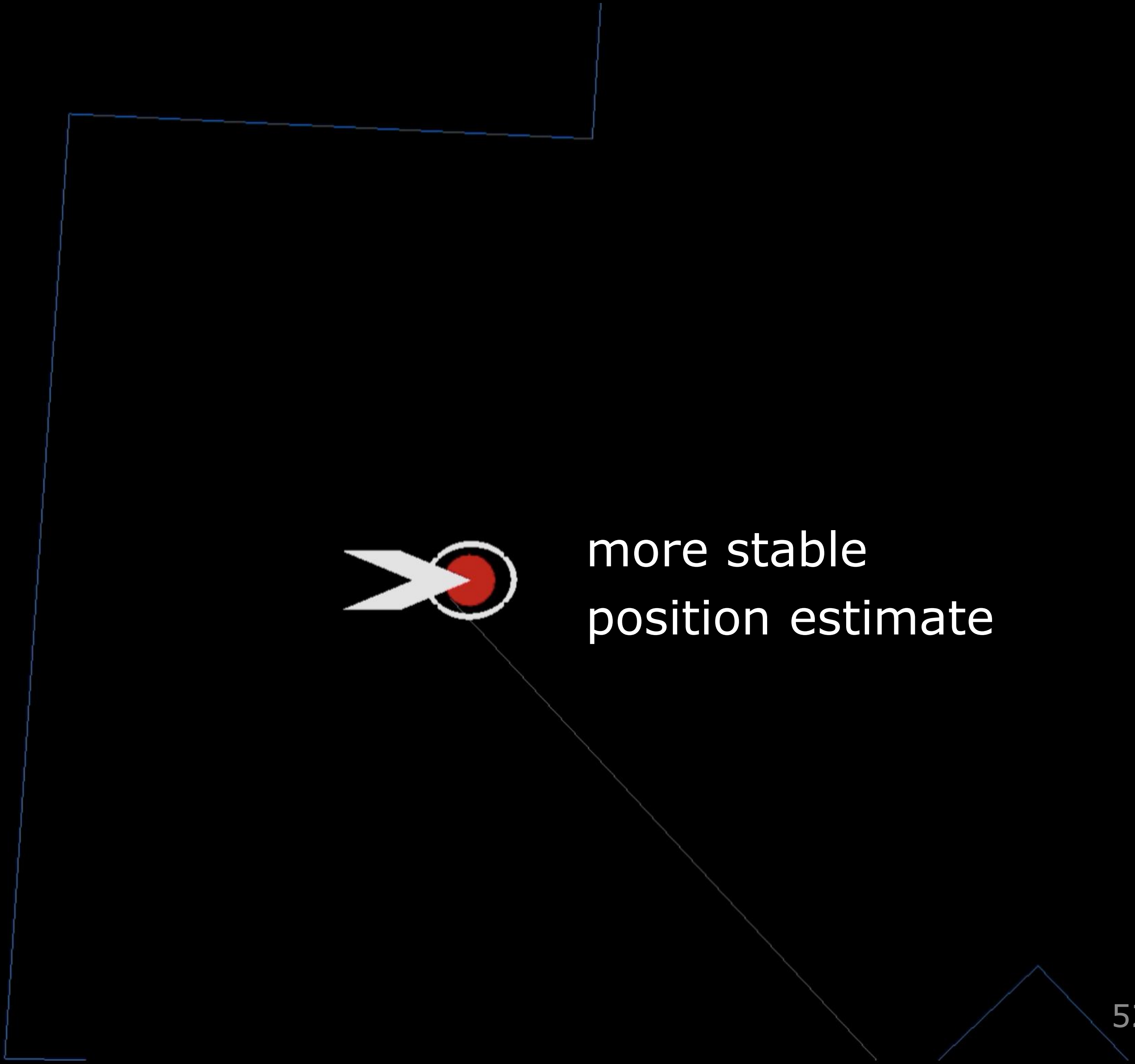
Sparse **global** priors

Eckenhoff et al. (2016):
1. optimize blanket
2. use local linearization

Wilbers et al. (2019):
1. use global linearization
2. compensate linearization errors

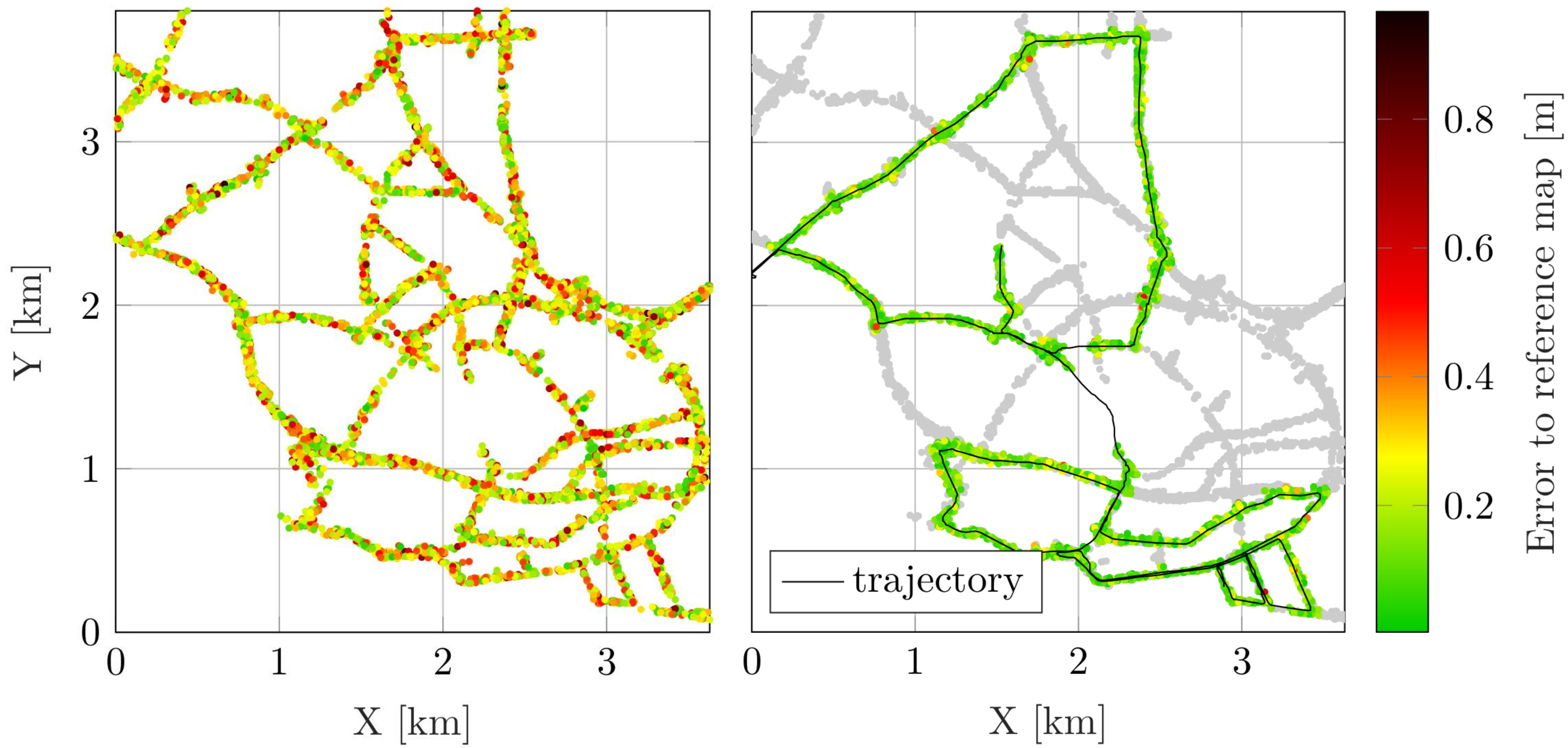
$$\tilde{p}(\boldsymbol{x}_n) = \prod_i \mathcal{N}\left(\breve{\boldsymbol{x}}_{n_i} \underline{- {\Omega_{a_i}}^{-1}\boldsymbol{b}_{t_i}}, {\Omega_{a_i}}^{-1}\right)$$





more stable
position estimate

Delayed map refinements



initial map → refined map

24.7 cm (avg. error) 12.3 cm (avg. error)

Summary

- **introduction** to the different facets
- using **landmarks** in urban scenarios
- data association for **third-party** maps
- **graph-based** sliding window optimization
- **estimating landmark refinements** in sliding window graphs