Deep Multi-Task Learning for Joint Localization, Perception, and Prediction

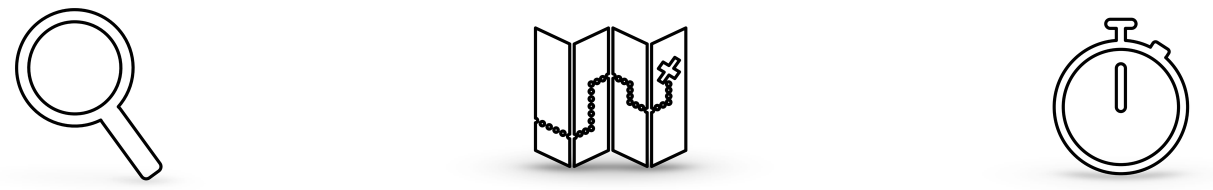
Uber ATG

John Phillips[1,2] Julieta Martinez[1] Ioan Andrei Bârsan[1,3] Sergio Casas[1,3] Abbas Sadat[1] Raquel Urtasun[1,3]
1) Uber ATG, 2) University of Waterloo, 3) University of Toronto

University of Toronto
University of Waterloo

1. Motivation

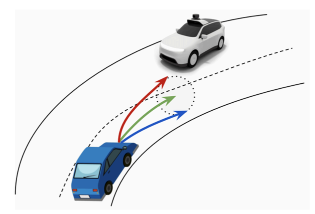
- Vehicle ego-localization is considered a key task for self-driving, as it enables the use of **HD maps**—strong priors for numerous tasks
- The effects of **localization error** on autonomy systems remain unquantified
- **Perception and prediction** (P2) rely on localization, but are usually performed independently after localization
- Learning-based localization methods are **robust** but computationally **expensive**
- We have three main goals:



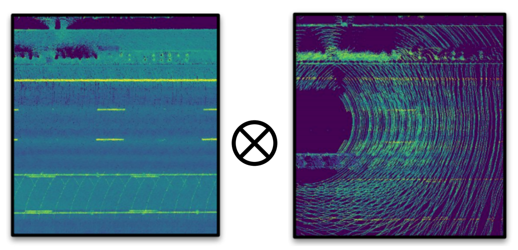
Quantify the effects of localization error on P2 | **Achieve high-accuracy localization** | **Share computation and reduce inference time**

2. Related work

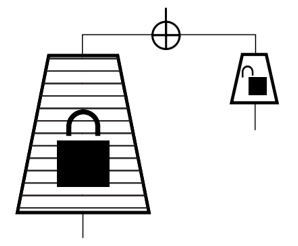
- **Learning-based Perception and Prediction (P2)**: LiDAR + high-definition maps to improve motion forecasting [Casas et al, '20]



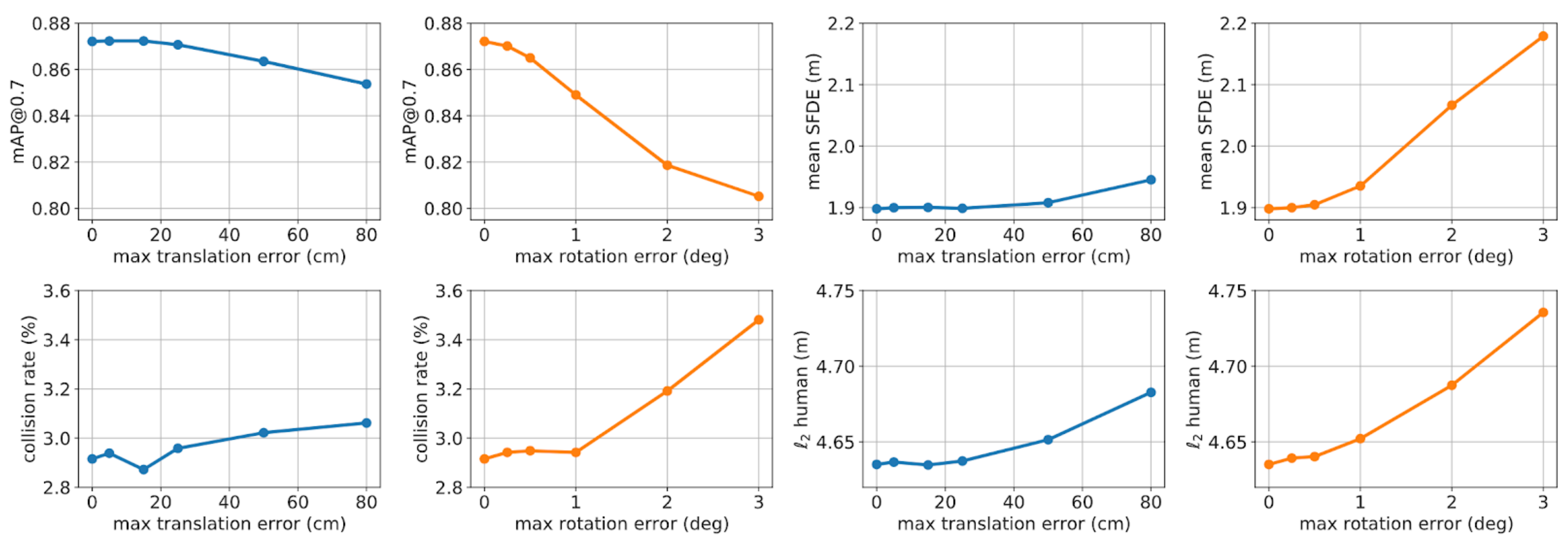
- ***Learning*-based *online* localization**: It is viable to cast localization as a learnable matching task [Bârsan et al, '18]



- **Multitask Learning**: We use **side-tuning**, which adds a small side-network to an existing network to reuse the features of a strong backbone in a new task, while avoiding catastrophic forgetting



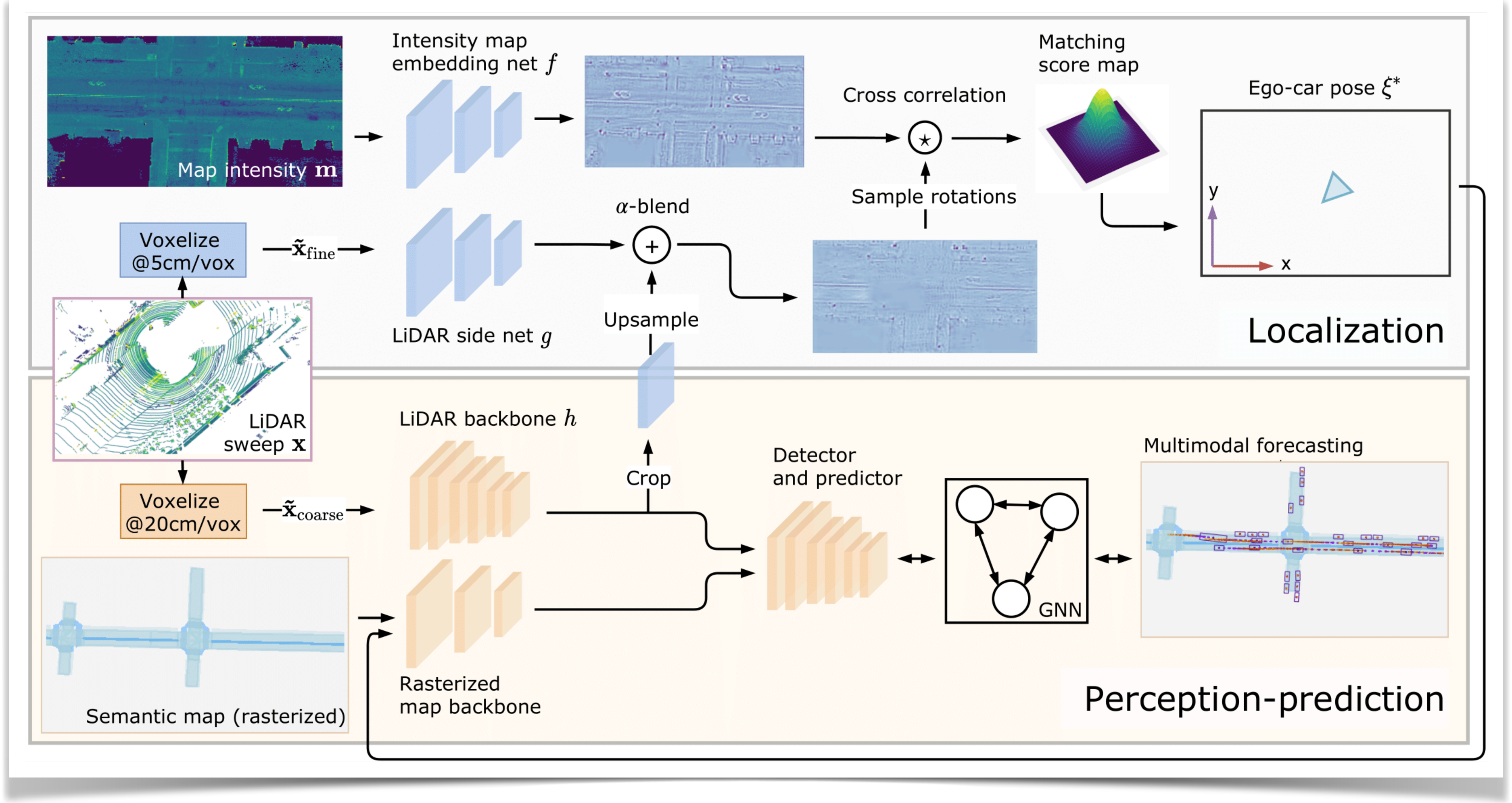
3. Localization error and autonomy



- **How do localization errors affect autonomy systems?**
- We systematically increase translational and rotational error, and evaluate the Implicit Latent Variable Model (ILVM) [Casas et al, '20] for P2 and the Path-Lateral Time (PLT) motion planner [Sadat et al, '19]
- Our results suggest that **small localization errors may be acceptable for autonomy**

4. Joint localization and P2

- Three main design considerations in mind for our joint localization, perception, and prediction (LP2) system:
  1. **Low latency**, and low overhead on top of the P2 system
  2. **Learning-based localization** for robustness against dynamic objects and LiDAR intensity miscalibration, and to re-use computation from the P2 side
  3. **Easier to train, evaluate, and deploy** than its separate counterparts



- **Localization:** We compute a matching score map using cross correlation between the LiDAR embedding $g(\tilde{\mathbf{x}}_{\text{fine}})$ and the map embedding $f(\mathbf{m})$:

$$\xi^* = \underset{\xi}{\operatorname{argmax}}\ \pi(g(\tilde{\mathbf{x}}_{\text{fine}}), \xi) \cdot f(\mathbf{m}) \triangleq \underset{\xi}{\operatorname{argmax}}\ \mathbf{p}(\xi),$$

where π is a function that warps its first argument based on the 3-DoF offset ξ

- The most likely pose is simply the argmax of the matching score maps
- The localization task is trained using the cross entropy loss between the matching score map and a one-hot encoded ground truth:

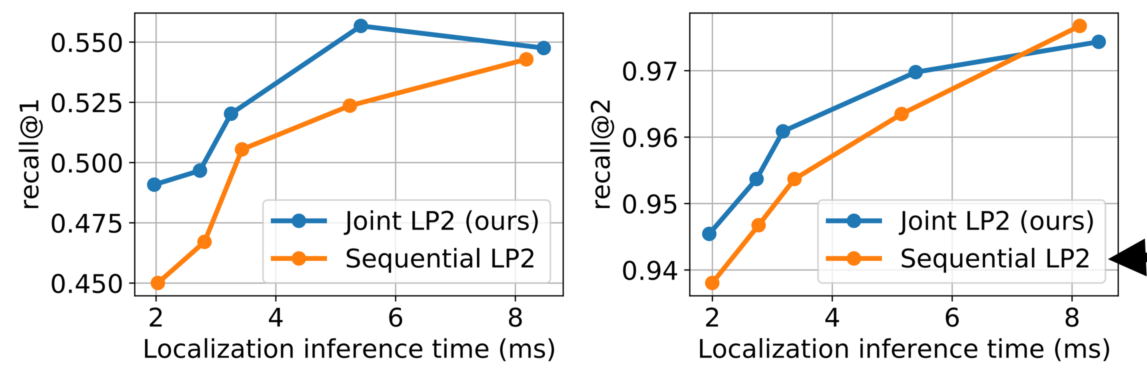
$$\mathcal{L}_{\text{Loc}} = -\sum_{\xi} \mathbf{p}(\xi)^{\text{GT}} \log \mathbf{p}(\xi)$$

- **Perception-prediction (P2):** We process voxelized LiDAR with a "heavy" backbone $h(\tilde{\mathbf{x}}_{\text{coarse}})$, and combine it with a lightweight processing of the rasterized semantic map to obtain dynamic objects and their future trajectories. This corresponds to the ILVM model using the pose computed by the localization module to rasterize the map
- **Optimization:** The localization module (upper part) is trained after freezing the perception-prediction model (bottom part) via side-tuning, which was necessary to avoid catastrophic forgetting
- **Feature sharing:** Importantly, we re-use upscaled features from the P2 module for the localization subsystem. This is crucial to achieve low latency

5. Results

- At different localization inference time budgets, our method is oftentimes more accurate in terms of localization error than running localization and P2 independently (sequentially)

Model	Time (ms)	recall@1	recall@2
LiDAR Localizer	25.92	0.52	0.95
LiDAR Localizer (Pixor-based)	2.79	0.47	0.95
Joint LP2 (Ours)	1.95	0.49	0.95



Separate LiDAR localization and perception

- Joint LP2 system benchmark, focusing on on motion planning:

Model	P2 pose (GT, N)	Planning Pose (GT, N)	r@1 ↑ (%)	r@2 ↑ (%)	Collision ↓ (% up to 5s)	L2 human ↓ (m @ 5s)
ILVM	GT	GT	-	-	**2.915**	**4.64**
ILVM	GT	N	-	-	3.168	4.68
ILVM	N	N	-	-	3.511	4.70
Joint LP2 — Ours (Tiny Pixor)	N	N	46.6	93.5	2.962	**4.64**
Joint LP2 — Ours (Big Pixor)	N	N	**52.5**	**96.9**	2.922	**4.64**

- The best results are achieved with an ILVM that has access to perfect (GT) localization — as it is often benchmarked — but this is unrealistic in practice
- We then **simulated noise** in the localizer (N) affecting either the P2 system, the motion planner, or both: this increases collision rates and the distance between the trajectory of the motion planned and that of an expert human driver
- Our method allows the system to relocalize, improving both key motion planning metrics. Although localization is not perfect, this is tolerable for autonomy as shown in our previous experiments

6. Conclusion and outlook

- We have investigated the effects of localization error on perception, prediction, and motion planning
- Localization can be accelerated dramatically by sharing computation with perception, while retaining accuracy—localization adds just 2ms of overhead!
- Several avenues for future work remain, including:
  - Evaluate performance in closed loop simulation
  - Thoroughly comparing classical localization approaches with this method
  - Further investigate localization failure cases by classifying collisions according to severity

Project Website