Uber ATG

We are hiring! Meet us at booth **#1155**.

Learning to Localize through Compressed Binary Maps

Xinkai Wei[1,2]*, Ioan Andrei Bârsan[1,3]*, Shenlong Wang[1,3]*, Julieta Martinez[1], Raquel Urtasun[1,3]

1) Uber ATG, 2) University of Waterloo, 3) University of Toronto

UNIVERSITY OF TORONTO

UNIVERSITY OF WATERLOO

1. Motivation

- **Centimeter-level localization** is a key task for self-driving.
- **Learning to match** observations to maps shown to be highly effective.
- Detailed maps can have very demanding **storage requirements**.
- **Goals**:

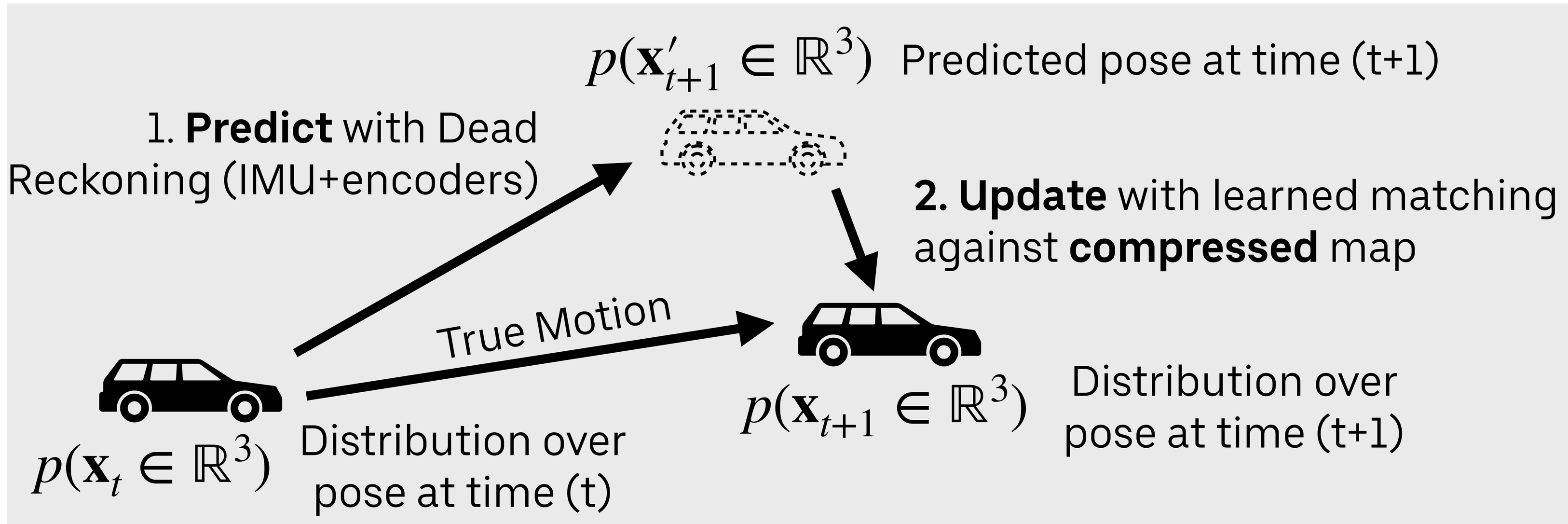
- Address his by **learning a compression scheme** optimal for localization by **jointly** learning localization and compression.

2. Related Work

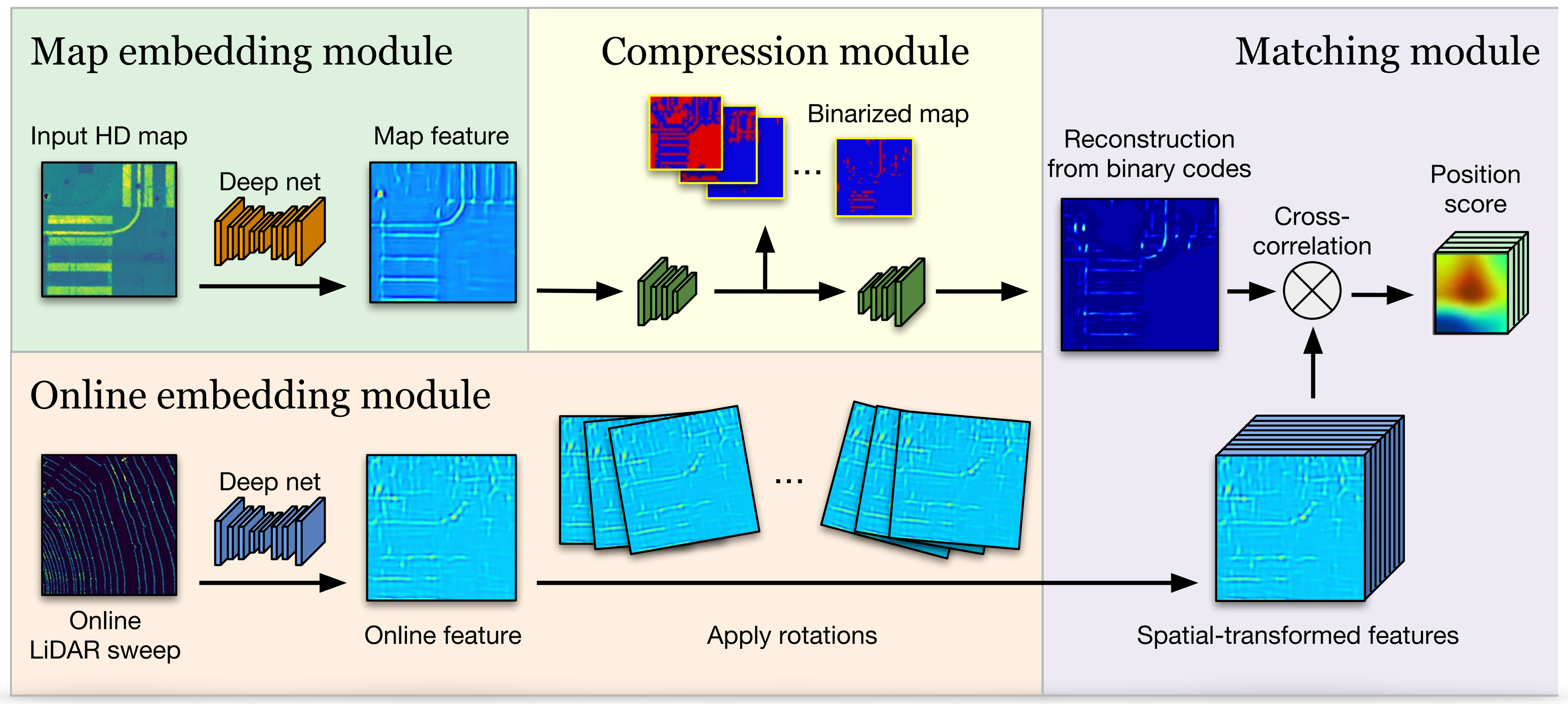
- **Learning-based Online Localization**
  - **Learning to Localize Using a LiDAR Intensity Map** (I. A. Bârsan et al., CoRL '18, our previous work) showed it is viable to cast localization as a learnable matching task.
  - **L^3-Net** by Lu et al., 2019 presents a system which learns to match point clouds for localization in an end-to-end pipeline.
- **Learning-based Image Compression**
  - **RNN-based** (Toderici et al, '15, '16', '17, etc.)
  - **GAN-based** (Rippel & Bourdev '17, Augustsson '18)
- **Task-specific compression** (videos, faces, medical imagery)

3. Probabilistic Localization

- Our goal is to perform **online localization**, and compute a centimeter-level accurate map-relative pose of the AV at every time step.
- The poses are parameterized with three degrees of freedom **(x, y, yaw)**.
- Localization follows a standard **histogram filtering** formulation.
- We train the matching module leveraged in the **update step** of the filter.
- This 3D search space is discretized, and searched exhaustively around the predicted pose at each time.
- **Predicted pose** = past pose + integrated IMU & wheel encoders.



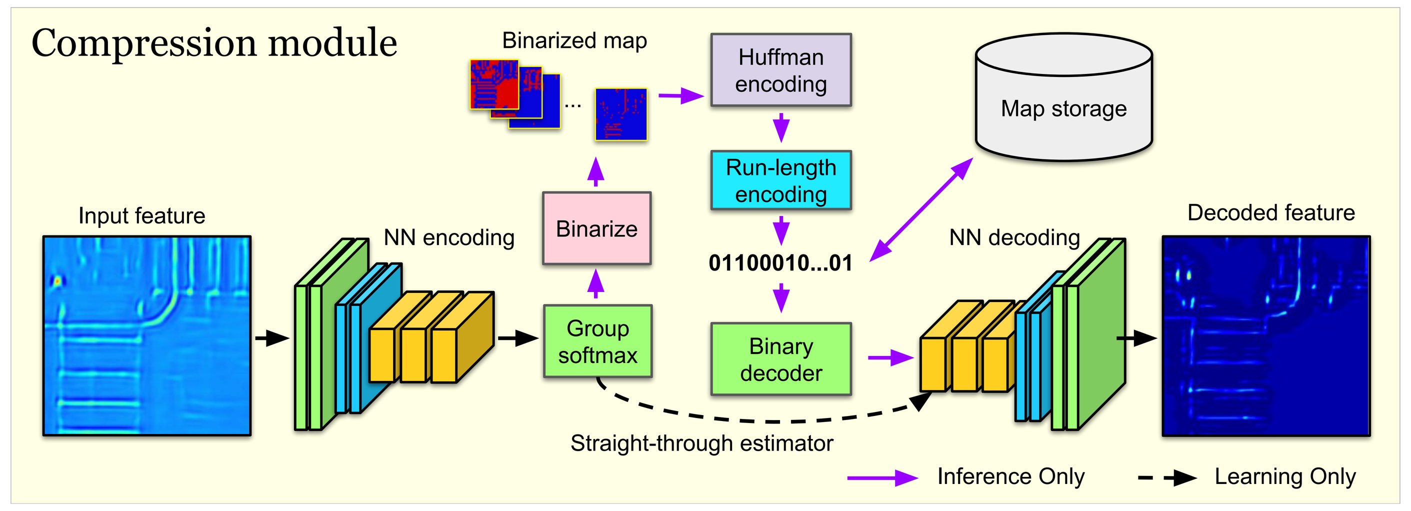
- The LiDAR matching depicted below is trained to match observations to **compressed maps**, using a learned matching method.



- Note: No need to compress online observations (never stored).
- Input LiDAR and maps are all in **bird's-eye view** (2D).

4. Learning to Compress & Match

- We compute feature embeddings for online data and for map data such that **matching** accuracy is maximized.
- Train with **compression in the loop** to reduce the map's bitrate.
- Build good **sparse binary** representations such that **Huffman** and **Run-Length Encoding** can do a very good job.



- Training to (1) maximize matching performance while (2) minimizing code length and (3) ensuring the **binarization-induced** error is minimal.

$$\ell = \ell_{\text{Loc}}(\mathbf{y}, \mathbf{y}_{\text{GT}}) + \lambda_1 \ell_{\text{CodeLen}}(\mathbf{p}) + \lambda_2 \ell_{\text{HardBin}}(\mathbf{p})$$

(1) Localization term: Cross-entropy between predicted 3D (x, y, yaw) score map and ground truth one-hot offset.

$$\ell_{\text{Loc}}(\mathbf{y}, \mathbf{y}_{\text{GT}}) = \sum_i y_{\text{GT},i} \log(y_i)$$

Output of Matching Pipeline

(2) Entropy in the mini batch is a differentiable surrogate of code length.

$$\ell_{\text{CodeLen}}(\mathbf{p}) = \bar{\mathbf{p}} \log \bar{\mathbf{p}} \qquad \bar{\mathbf{p}} = \frac{1}{W \times H \times B} \sum_i \mathbf{p}_i$$

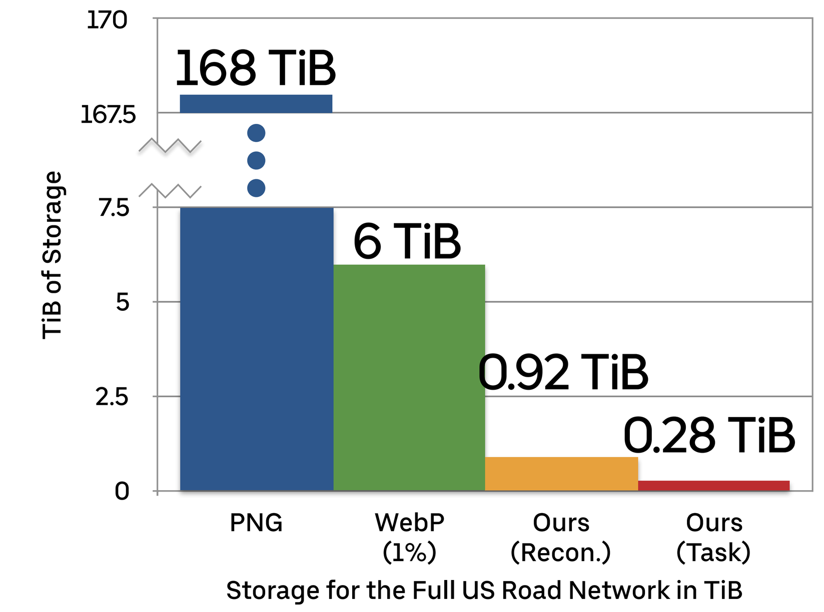
(3) Minimize **per-pixel** entropy to reduce **hard binarization-induced** error.

$$\ell_{\text{HardBin}}(\mathbf{p}) = \sum_i p_i \log p_i$$

Output of Group Softmax

5. Results

- **200x** better than lossless.
- **20x** better than lowest-quality lossy WebP codec.
- **40%** better than generic learning baseline.
- Results enable maps of country-wide road networks to fit onboard storage.
- Regional maps can fit in RAM.



Method	Median error (cm) Lat	Lon	Total	Failure rate (%) ≤ 100m	≤ 500m	End	Bit per pixel
Lossless (PNG)	1.55	**2.05**	**3.09**	**0.00**	1.09	2.44	4.94
JPG-5	4.32	5.48	8.41	**0.00**	1.09	1.25	0.18
JPG-50	3.29	5.60	7.59	**0.00**	1.09	5.26	1.03
WebP-5	1.65	5.75	6.53	2.04	5.43	13.95	0.30
WebP-50	1.62	2.75	3.76	**0.00**	3.26	3.30	1.05
Ours	1.61	2.26	3.47	**0.00**	1.09	**1.22**	**0.0083**

Comparison to **non-learning** baselines on our **urban** dataset.

Method	Median error (cm) Lat	Lon	Total	Failure rate (%) ≤ 100m	≤ 500m	End	Bit per pixel
Lossless (PNG)	**1.55**	**2.05**	**3.09**	**0.00**	1.09	2.44	4.93580
Ours (recon, 8×)	1.59	2.16	3.24	**0.00**	1.09	**1.22**	0.02689
Ours (recon, 16×)	1.76	2.48	3.62	**0.00**	**0.00**	2.56	0.01155
Ours (match, 8×)	1.61	2.26	3.47	**0.00**	1.09	**1.22**	0.00830
Ours (match, 16×)	1.62	2.77	3.84	1.00	2.17	4.26	**0.00733**

Comparison to **learning-based** baselines on our **urban** dataset.

Method	Median Err (cm) Lat	Lon	Total	Failure Rate (%) ≤ 100m	≤500m	End	b/m²
PNG, 5cm/px	**1.55**	**2.05**	**3.09**	**0.00**	1.09	**2.44**	1948.55
PNG, 10cm/px	4.37	6.68	9.50	3.19	3.26	4.00	402.84
JPG@50, 10cm/px	4.51	5.78	8.95	0.00	1.09	10.64	63.42
PNG, 15cm/px	15.73	23.66	31.73	10.31	20.65	22.03	173.97
JPG@50, 15cm/px	11.67	18.20	25.14	9.28	13.04	16.28	29.00
Ours (16×)	1.76	2.48	3.62	**0.00**	**0.00**	2.56	**2.87**

Ablation: Error, Failure Rate and bits/m² as a function of **map resolution** (cm/px).

6. Conclusions & Outlook

- This work addresses one of the main challenges associated with high-definition maps: storage.
- We've shown that task-specific compression can improve over general-purpose compression, allowing giant maps to be kept in-memory.
- Several avenues for future work remain, including:
  - Investigating methods for compressing 3D point clouds and doing full **six-degrees-of-freedom** localization.
  - End-to-end learning with the pose filter in the loop, similar to L^3-Net.
  - Learning with **mapping-in-the-loop**.