Towards End-to-End GPS Localization with Neural Pseudorange Correction

Xu Weng[1], KV Ling[1], Haochen Liu[1], Kun Cao[2]

[1]Nanyang Technological University

[2]KTH Royal Institute of Technology

July 10, 2024

Background (1/4)

- **High-performance Positioning/Localization Equipment:**

NovAtel's leading SPAN® GNSS+INS Technology

PwrPak7-E1
from HEXAGON

VEXXIS® GNSS-800 Series
Antennas from HEXAGON

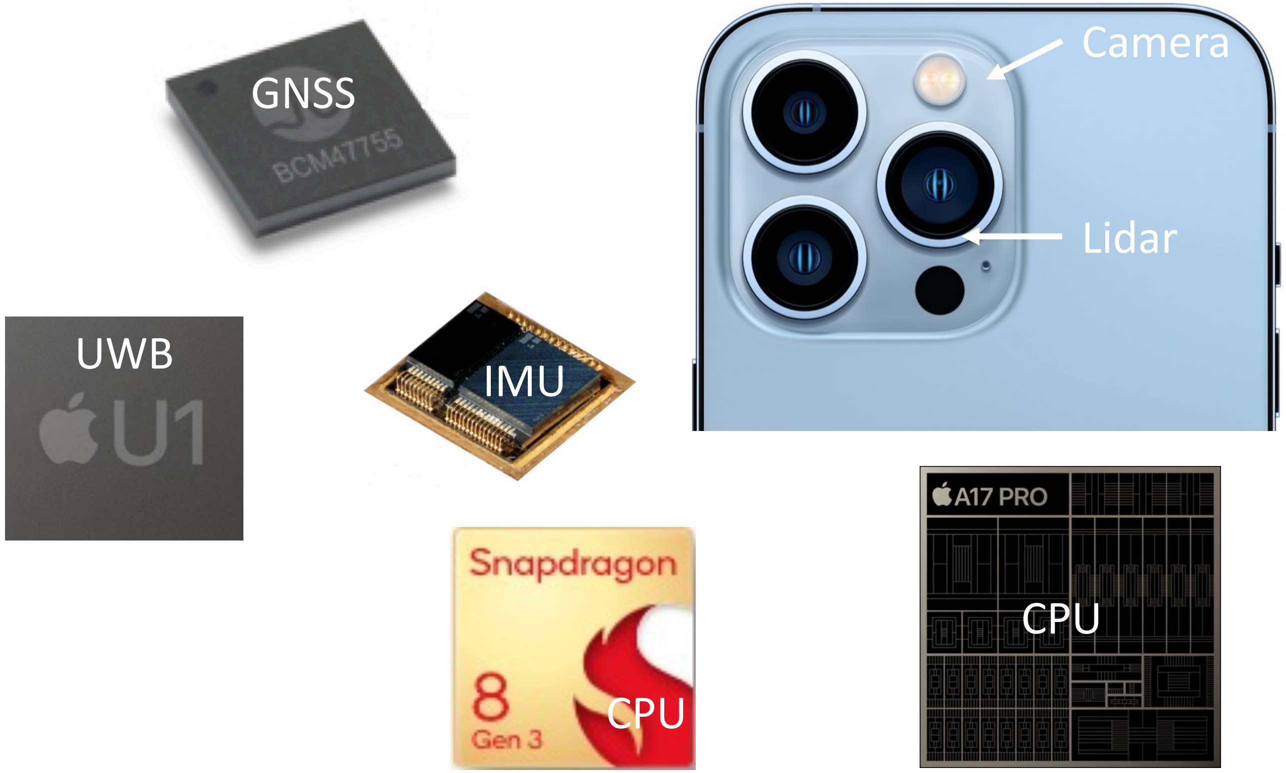
Waymo's Integrated System of Sensors

Lidar + Camera + Radar
(From YouTube)

- **Inaccessibility in daily life**
  - *Heavy*
  - *Large*
  - *Expensive*

Background (2/4)

- By contrast, smartphones are:
  - ➢ *Affordable*
  - ➢ *Portable*
  - ➢ *Integrated with other sensors*



Navigation

Bike Sharing

Emergency Location Service

Mobile AR

(From Google Images)

Background (3/4)

GNSS Satellite

GNSS Signals

Android Smartphone

GnssLogger App

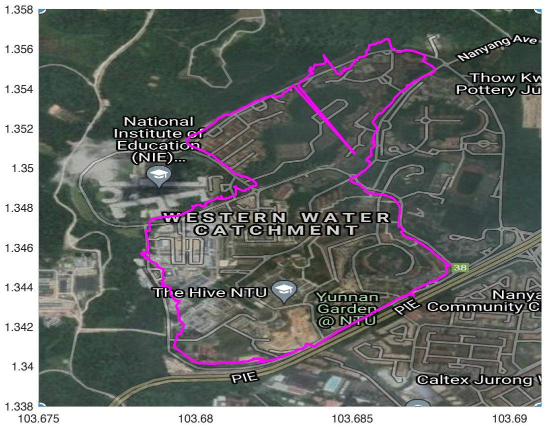
Developed with Google

GnssLogger

Log GNSS Data

GNSS Data File

Analysis Software



Background (4/4)

- In this work, we focus on the pseudorange-based localization solutions.

```
• Clock Field
    'TimeNanos'
    'TimeUncertaintyNanos'
    'LeapSecond'
    'FullBiasNanos'
    ......
• Measurement Field
    'Cn0DbHz'
    'ConstellationType'
    'MultipathIndicator'
    'PseudorangeRateMetersPerSecond'
    ......
```

Android Raw GNSS Measurements

The n^{th} satellite at the k^{th} time step:

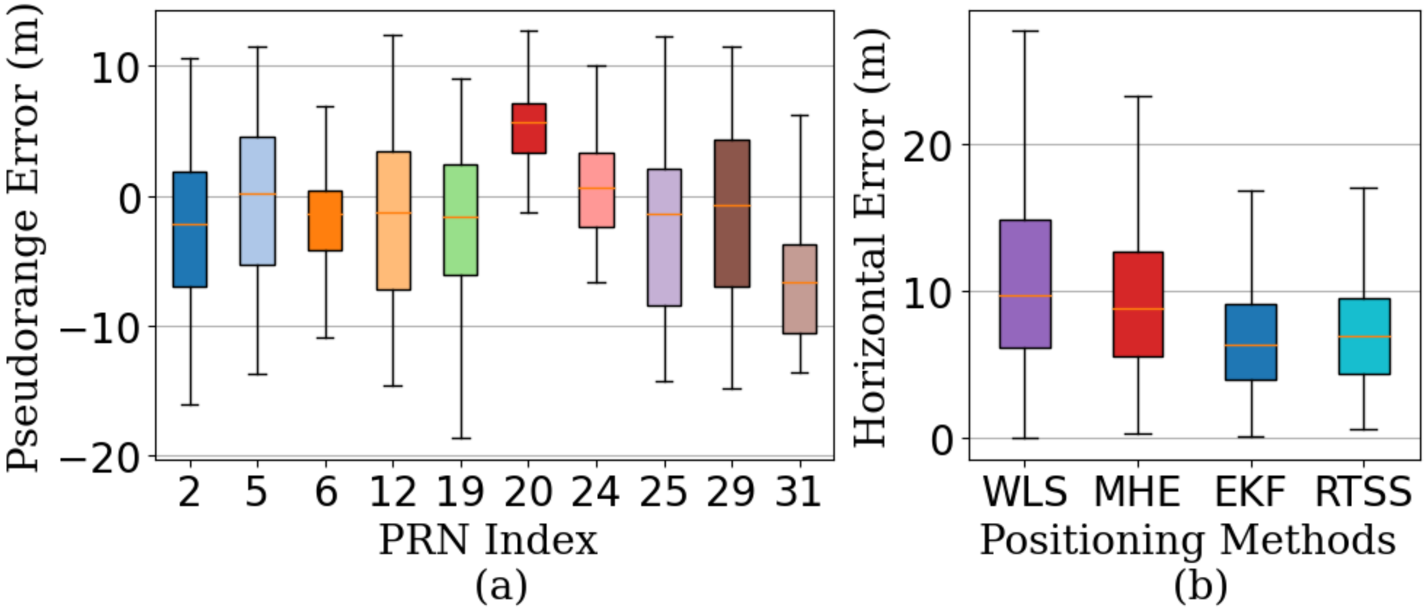
Pseudorange $$\rho_k^{(n)} = \underbrace{r_k^{(n)}}_{\text{Geometry distance}} + \underbrace{\delta t_{u_k}}_{\text{User's clock offset}} + \underbrace{\varepsilon_k^{(n)}}_{\text{Pseudorange noise and errors}}$$

We have modeled and removed:

1. Satellite clock errors
2. Relativistic effect
3. Group delays
4. Ionospheric delays
5. Tropospheric delays

Motivation (1/3)

- GNSS measurements from smartphones have:
  - ***Pseudorange errors, impacting positioning errors as a result***
  - The errors might be caused by multipath, non-line-of-sight (NLOS) propagation, modeling residuals, smartphone hardware delays, etc.



Trace "2021-04-28-US-MTV-1" collected by Pixel4 in GSDC dataset[1]

(a) Pseudorange errors of all visible satellites throughout the trace.

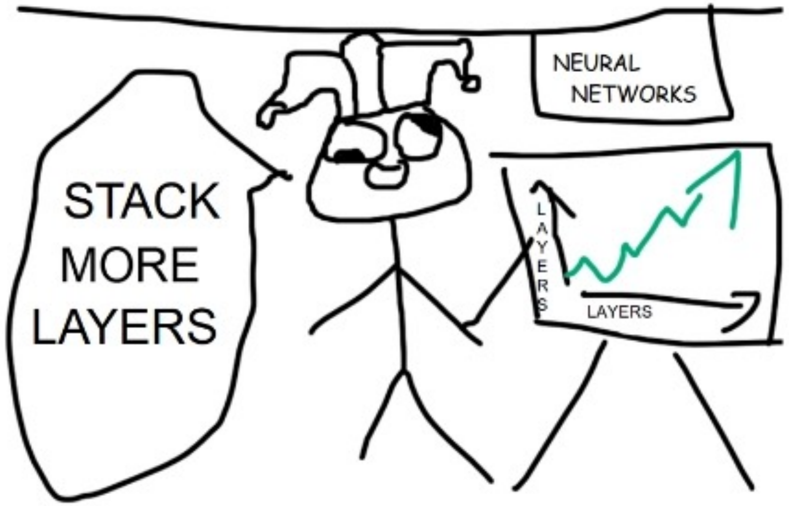
(b) Horizontal errors calculated with Vincenty's formulae using Weighted Least Squares (WLS), Moving Horizon Estimation (MHE), Extended Kalman Filter (EKF), and Rauch-Tung-Striebel Smoother (RTSS).

[1] X. Weng, K. Ling and H. Liu (2024), "***PrNet: A Neural Network for Correcting Pseudoranges to Improve Positioning With Android Raw GNSS Measurements***," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2024.3392302.

Correcting Pseudoranges

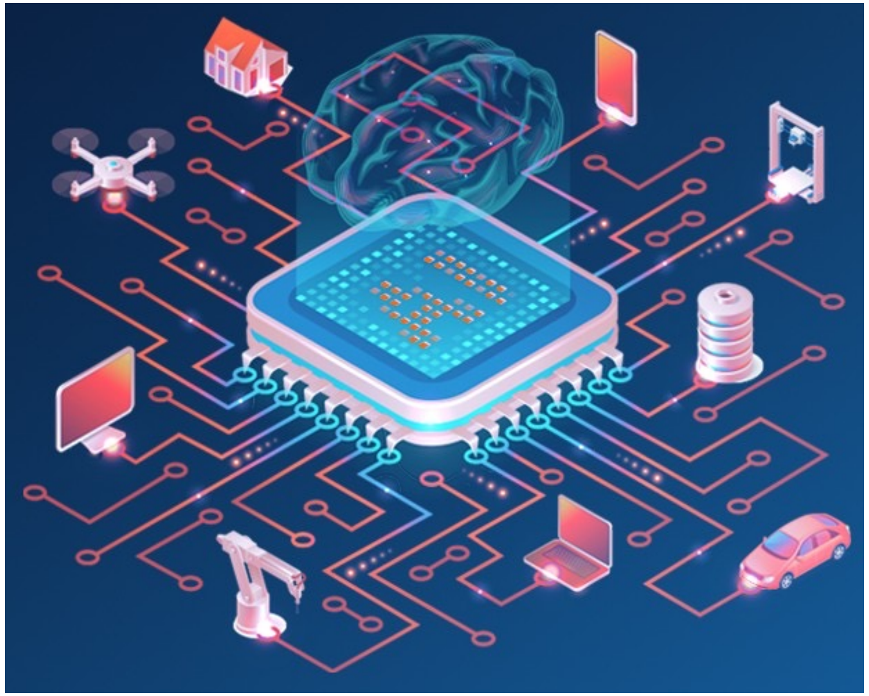
From Model-based to Data-driven Methods

From Google Images



From Google Images

The High Cost of Labeling Training Data in CPS/IoT Tasks (Embedded AI)

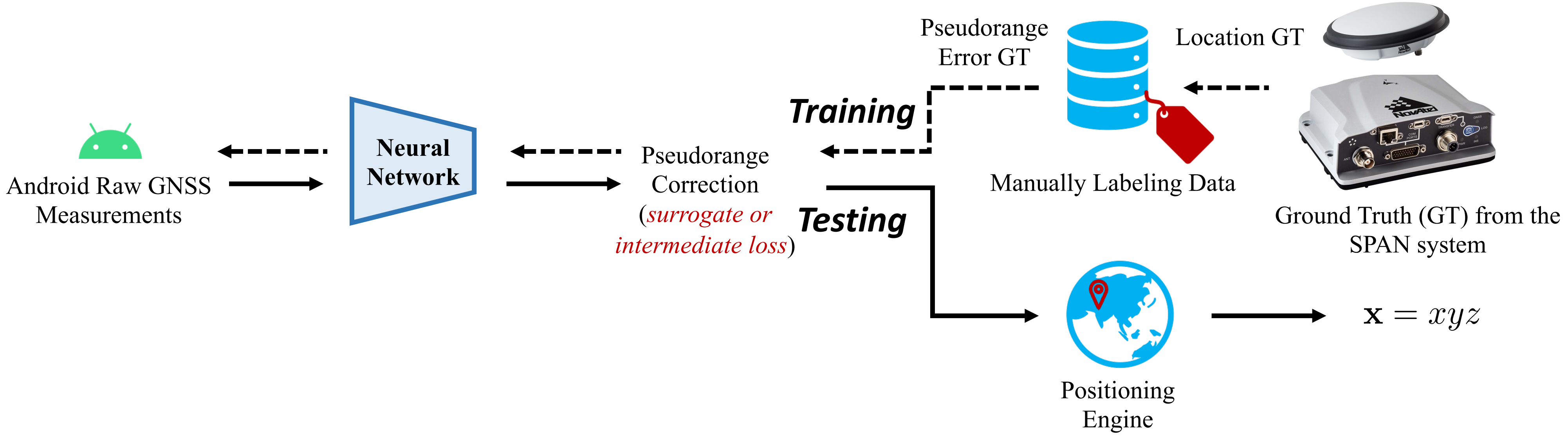


From Google Images

From Google Images

Related Work (1/3)

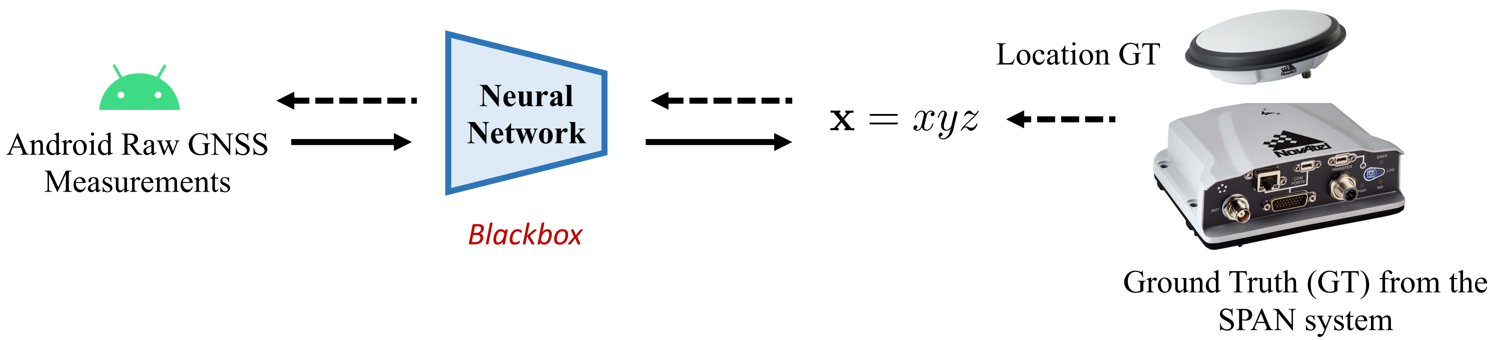
- Neural pseudorange correction uses *surrogate or intermediate* loss.
  - FCNN-LSTM[1]: double-differentiated pseudorange errors (with noise)
  - PBC-RF[2]: pseudorange errors (with noise)
  - PrNet[3]: smoothed pseudorange errors (filtered noise)



[1] Zhang, G., Xu, P., Xu, H., & Hsu, L. T. (2021). Prediction on the urban GNSS measurement uncertainty based on deep learning networks with long short-term memory. IEEE Sensors Journal, 21(18), 20563-20577.
[2] Rui Sun, Linxia Fu, Qi Cheng, Kai-Wei Chiang, and Wu Chen. Resilient pseudorange error prediction and correction for GNSS positioning in urban areas. IEEE Internet of Things Journal, 2023.
[3] X. Weng, K. Ling and H. Liu (2024), "PrNet: A Neural Network for Correcting Pseudoranges to Improve Positioning With Android Raw GNSS Measurements," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2024.3392302.

Related Work (2/3)

- End-to-end learning for GNSS localization *without using a model*.
  - Set Transformer[1]: replaces classical positioning engines with set transformer
  - Graph Convolution Neural Network (GCNN) [2]: replaces classical positioning engines with GCNN



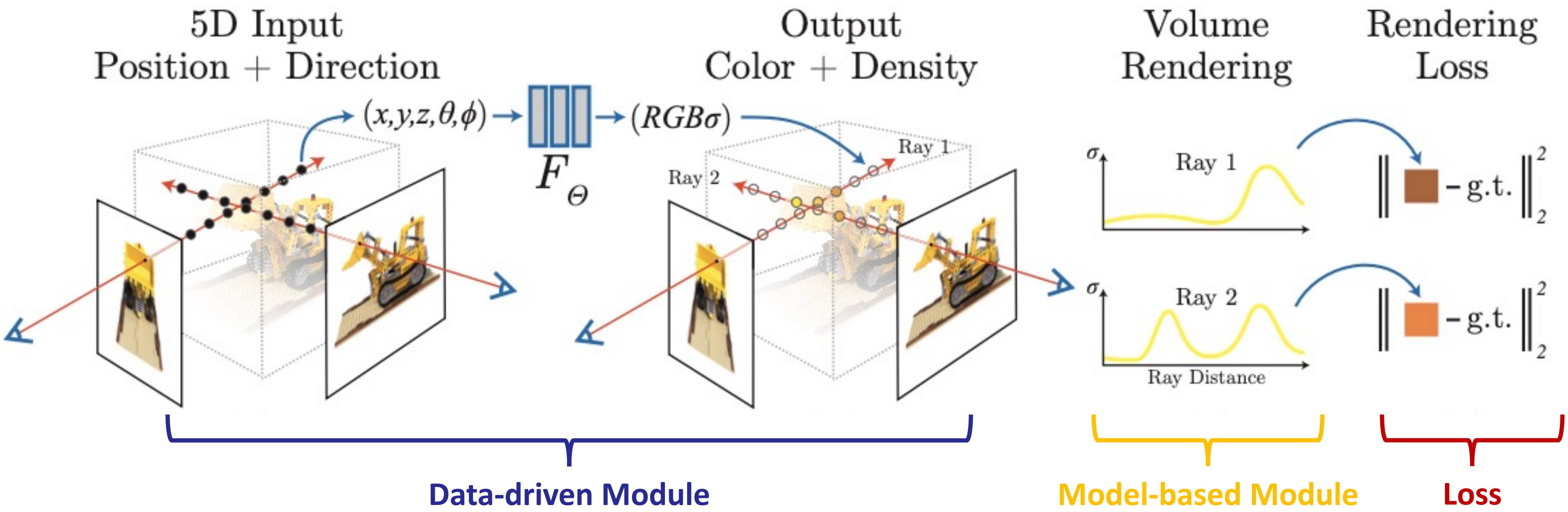
[1] Kanhere, A. V., Gupta, S., Shetty, A., & Gao, G. (2022). Improving GNSS positioning using neural-network-based corrections. NAVIGATION: Journal of the Institute of Navigation, 69(4).

[2] Mohanty, A., & Gao, G. (2023). Learning GNSS positioning corrections for smartphones using graph convolution neural networks. NAVIGATION: Journal of the Institute of Navigation, 70(4).

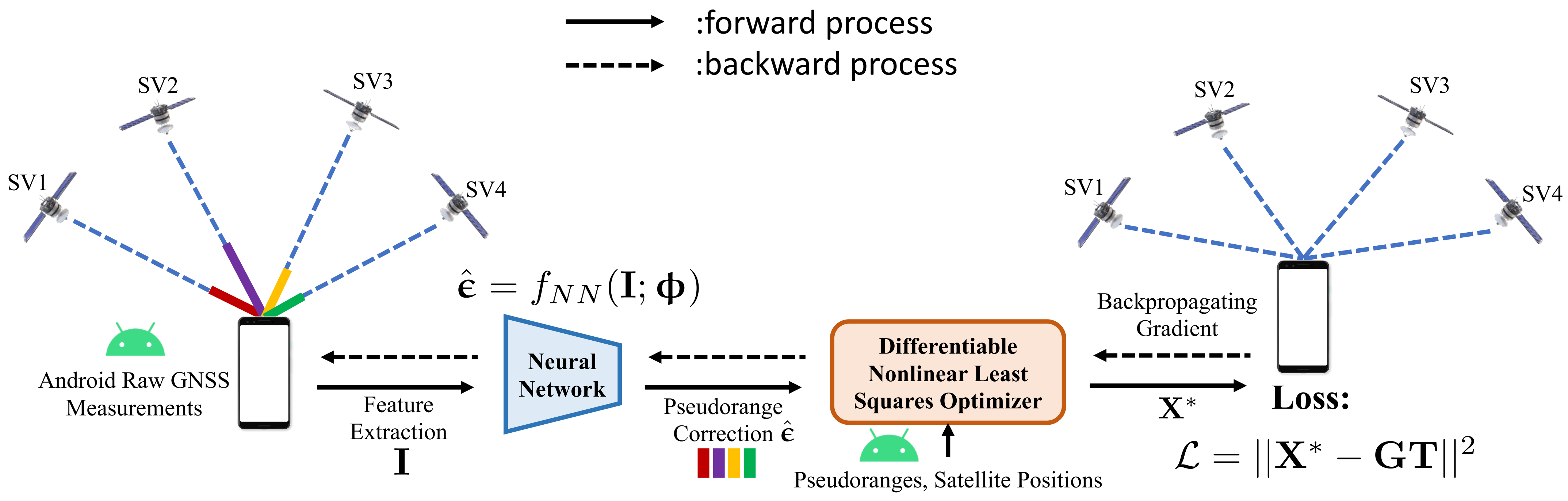
Related Work (3/3)

- Can we *combine Data-driven and Model-based approaches* so that they can cooperate and be in tune with one another?
  - Successful examples in other domains: NeRF (ECCV, 2020 Best Paper), EPro-PnP (CVPR, 2022 Best Student Paper), etc.

The Pipeline of Neural Radiance Fields



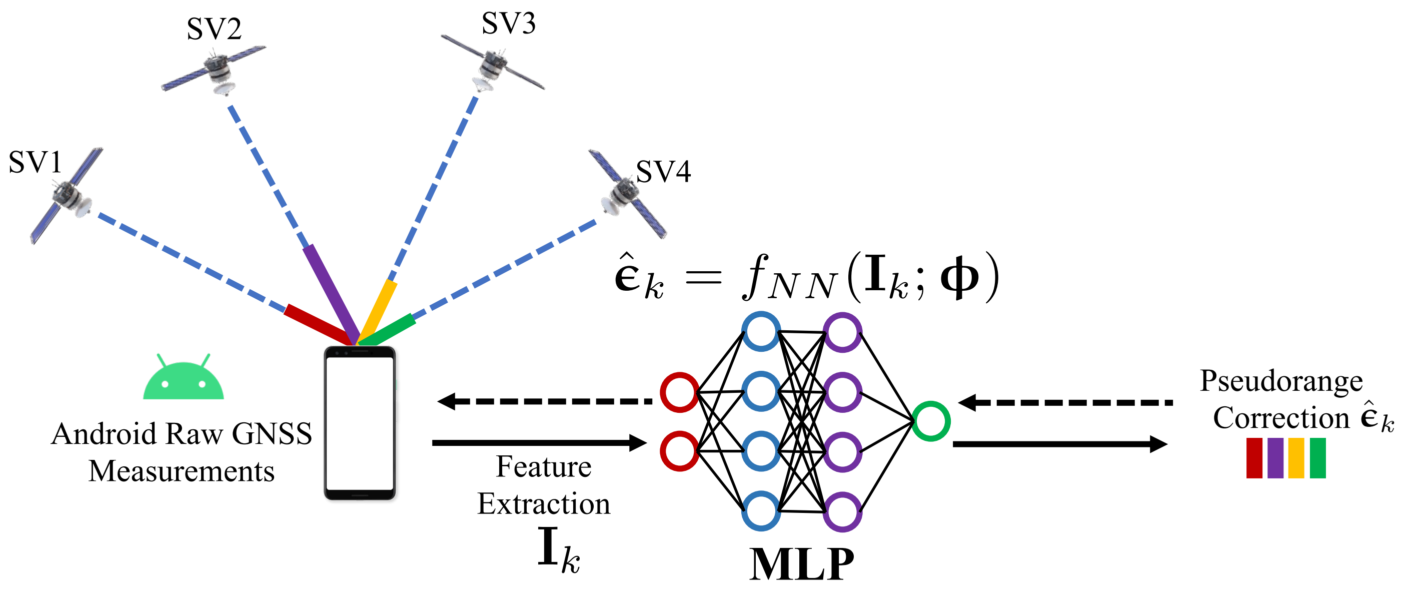
Our Proposed Framework: E2E-PrNet



- End-to-End Neural Pseudorange Correction (E2E-PrNet) consists of
  - The front-end neural network for pseudorange correction (PrNet)
  - The model-based differentiable nonlinear least squares (DNLS) optimizer
- Pseudorange correction is learned via minimizing positioning errors.

E2E-PrNet: the Front-end PrNet

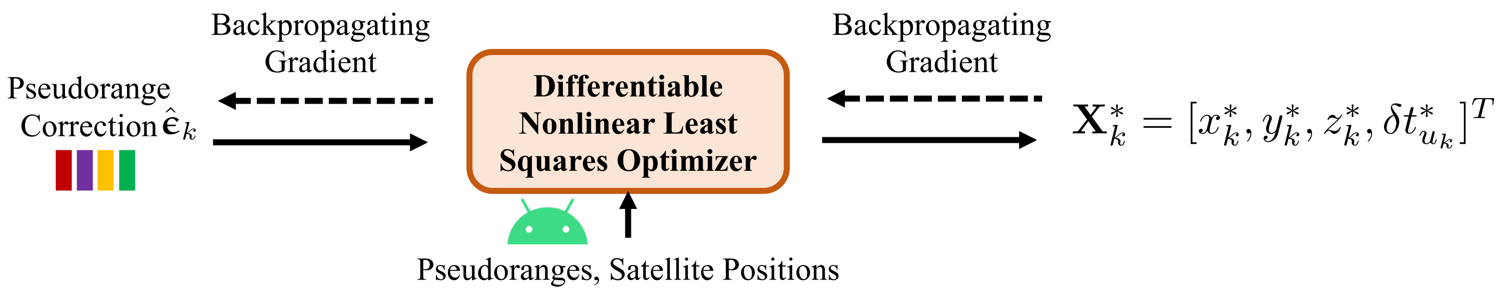
- We use PrNet as the front-end neural network.
  - Input feature vector: $\mathbf{I}_k$
  - Learnable weights of PrNet: $\boldsymbol{\phi}$
  - Pseudorange correction: $\hat{\boldsymbol{\epsilon}}_k$
  - The mapping performed by PrNet: $\hat{\boldsymbol{\epsilon}}_k = f_{NN}(\mathbf{I}_k; \boldsymbol{\phi})$



The front-end PrNet

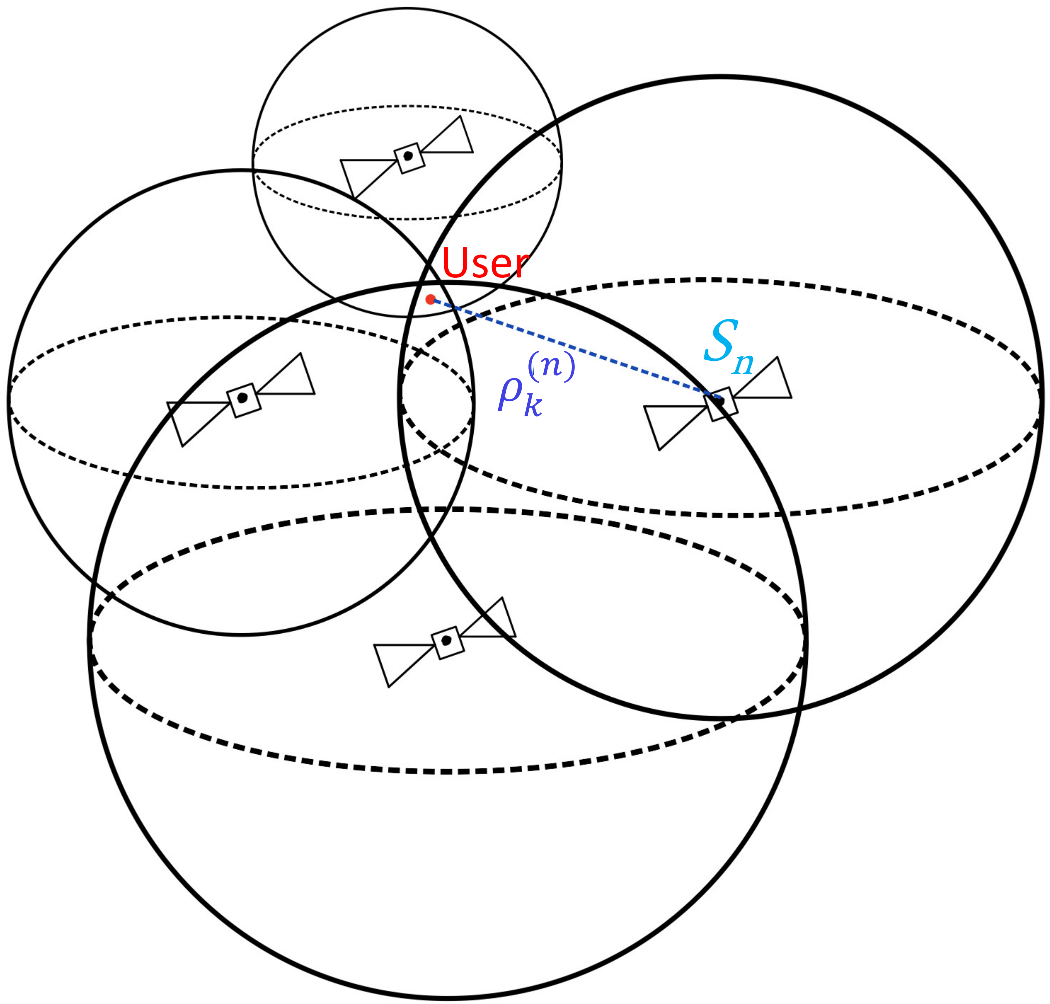
E2E-PrNet: the Back-end DNLS Optimizer (1/3)

- GNSS localization is formulated as a nonlinear least squares optimization problem.
  - Input of the DNLS optimizer:
    - Satellite positions: $[\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \cdots, \mathbf{x}_k^{(M)}]^T$
    - Pseudorange measurements: $[\rho_k^{(1)}, \rho_k^{(2)}, \cdots, \rho_k^{(M)}]^T$
    - Pseudorange correction from the front-end PrNet: $\hat{\boldsymbol{\epsilon}}_k = [\hat{\varepsilon}_k^{(1)}, \hat{\varepsilon}_k^{(2)}, ..., \hat{\varepsilon}_k^{(M)}]^T$
  - Output of the DNLS optimizer:
    - Optimal user location estimates: $x_k{}^*, y_k{}^*, z_k{}^*$
    - Optimal user receiver clock offset estimate: $\delta t^*_{u_k}$



The back-end DNLS Optimizer

E2E-PrNet: the Back-end DNLS Optimizer (2/3)



- We can establish M nonlinear equations using pseudorange measurements to calculate positions and clock offsets:

$$\sqrt{\left(x_k - x_k^{(1)}\right)^2 + \left(y_k - y_k^{(1)}\right)^2 + \left(z_k - z_k^{(1)}\right)^2} + \delta t_{u_k} = \rho_k^{(1)}$$
$$\sqrt{\left(x_k - x_k^{(2)}\right)^2 + \left(y_k - y_k^{(2)}\right)^2 + \left(z_k - z_k^{(2)}\right)^2} + \delta t_{u_k} = \rho_k^{(2)}$$
$$\vdots$$
$$\sqrt{\left(x_k - x_k^{(M)}\right)^2 + \left(y_k - y_k^{(M)}\right)^2 + \left(z_k - z_k^{(M)}\right)^2} + \delta t_{u_k} = \rho_k^{(M)}$$

E2E-PrNet: the Back-end DNLS Optimizer (3/3)

- Problem Formulation:
  - Correct pseudorange measurements using the output of the front-end PrNet:

$$\rho_{c_k}^{(n)} = \rho_k^{(n)} - \hat{\varepsilon}_k^{(n)}$$

  - Recall the pseudorange equation for the n^{th} satellite at time k:

$$\sqrt{(x_k - x_k^{(n)})^2 + (y_k - y_k^{(n)})^2 + (z_k - z_k^{(n)})^2} + \delta t_{u_k} = \rho_{c_k}^{(n)}$$

  - The objective is written as:

$$\mathbf{X}_k^* = \min_{\mathbf{X}_k = [x_k, y_k, z_k, \delta t_{u_k}]^T} S(\mathbf{X}_k)$$

where

$$S(\mathbf{X}_k) = \frac{1}{2}||\mathbf{l}_k(\mathbf{X}_k)||^2$$

$$\mathbf{l}(\mathbf{X}_k) = \left[l_k^{(1)}, l_k^{(2)}, ..., l_k^{(M)}\right]^T$$

$$l_k^{(n)} = (\rho_k^{(n)} - \hat{\varepsilon}_k^{(n)}) - (\sqrt{(x_k - x_k^{(n)})^2 + (y_k - y_k^{(n)})^2 + (z_k - z_k^{(n)})^2} + \delta t_{u_k}).$$

Loss Function

- We use the squared loss function:

$$\mathcal{L} = ||\mathbf{X}_k^* - \mathbf{X}_{GT_k}||^2$$

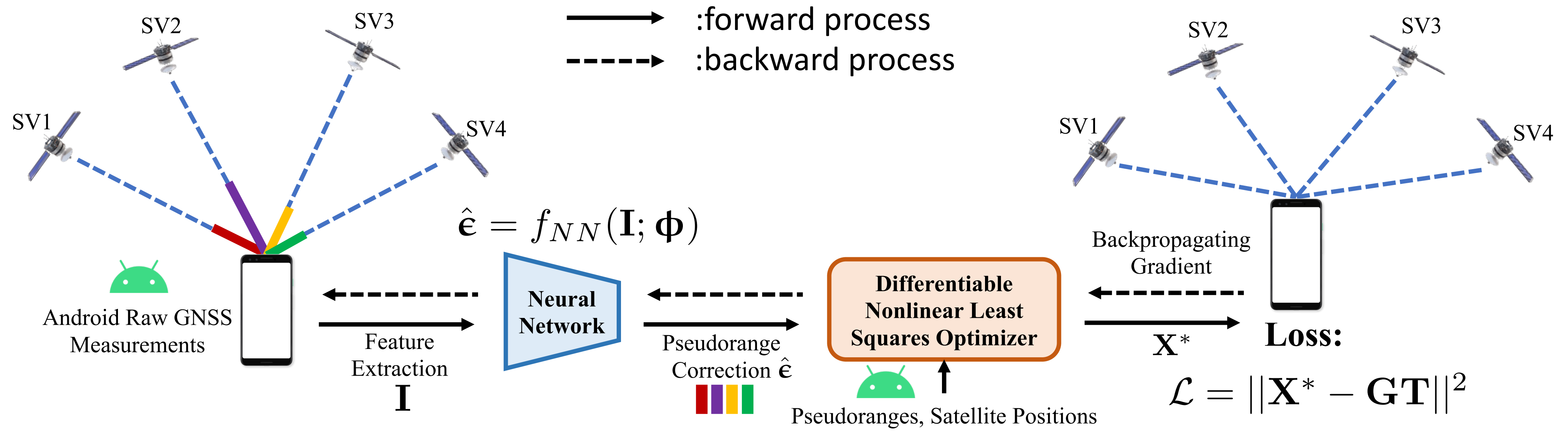
where $\mathbf{X}_{GT_k} = [x_{GT_k}, y_{GT_k}, z_{GT_k}, \delta t_{u_{GT_k}}]^T$ is the ground truth of user location and clock offset.

- The ground truth location can be obtained using high-performance geodetic GNSS receivers.
- However, the user clock offset is hardly labeled.
  - ✓We use its Weighted Least Squares (WLS)-based estimate to label it, i.e.,

$$\delta t_{u_{GT_k}} \leftarrow \delta \hat{t}_{u_k}^{(\mathrm{WLS})}.$$

Training E2E-PrNet (1/2)

- When we train E2E-PrNet, the gradient backpropagation is performed as



the first derivative is easy to compute considering its explicit form.

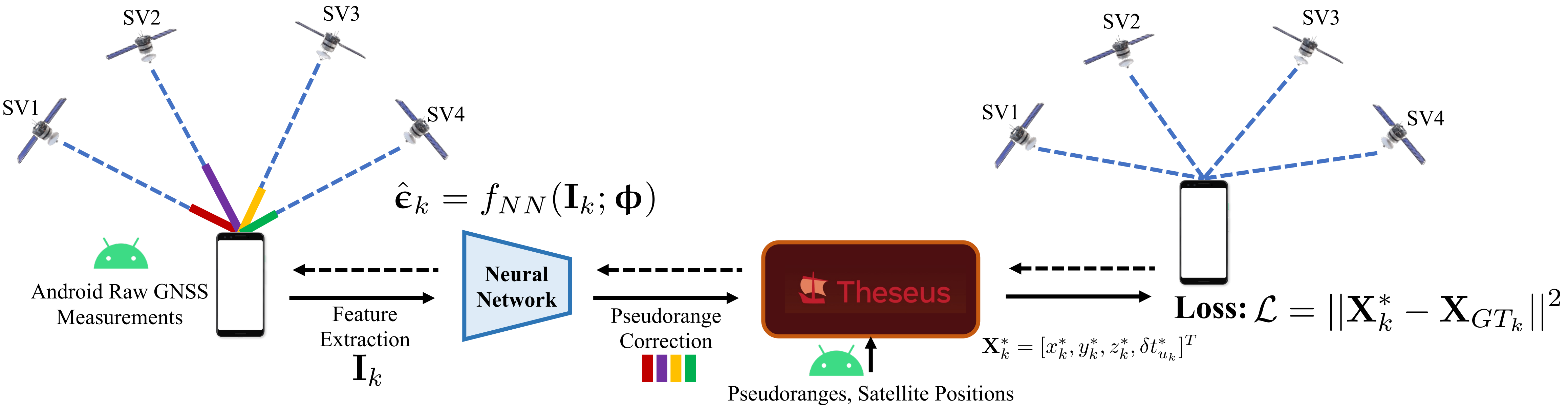
$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}} = \boxed{\frac{\partial \mathcal{L}}{\partial \mathbf{X}_k^*}} \cdot \boxed{\frac{\partial \mathbf{X}_k^*}{\partial \hat{\boldsymbol{\epsilon}}_k}} \cdot \boxed{\frac{\partial \hat{\boldsymbol{\epsilon}}_k}{\partial \boldsymbol{\phi}}}$$

the last one can be solved in a standard way by PyTorch or TensorFlow.

The differentiation through the nonlinear least squares optimization is difficult!!!

Training E2E-PrNet (2/2)

- We use Theseus, a general *differentiable nonlinear least squares solver from Meta AI*, to *solve* the nonlinear least squares optimization problem and *backpropagate* gradients.



[1] Pineda, L., Fan, T., Monge, M., Venkataraman, S., Sodhi, P., Chen, R. T., ... & Mukadam, M. (2022). Theseus: A library for differentiable nonlinear optimization. Advances in Neural Information Processing Systems, 35, 3801-3818.

Experiments: Google Smartphone Decimeter Challenge (GSDC) Datasets

Training and testing data were collected ***along the same routes on different dates***.

Scenario I: rural fingerprinting

Scenario II: rural cross-trace

Training and testing data were collected ***along the different routes on different dates***.

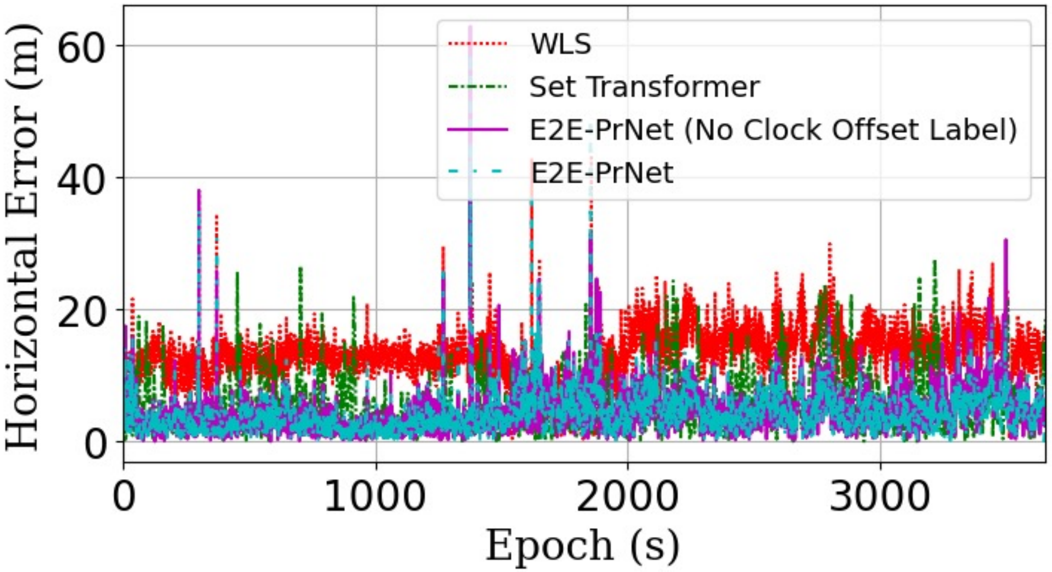
Details of Data Sets

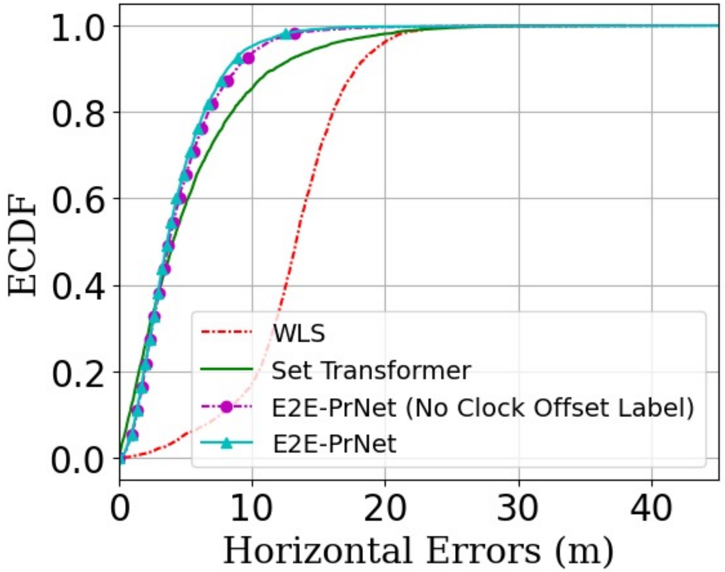
Scenarios	Time Length	Trace Distance	Smartphones	Urban Canyon
Training Data in Scenario I & II	5.5 h	650 km	Pixel 4	Light
Testing Data in Scenario I	1 h	120 km	Pixel 4	Light
Testing Data in Scenario II	0.5 h	11 km	Pixel 4	Medium

Experiments: Evaluation Metrics

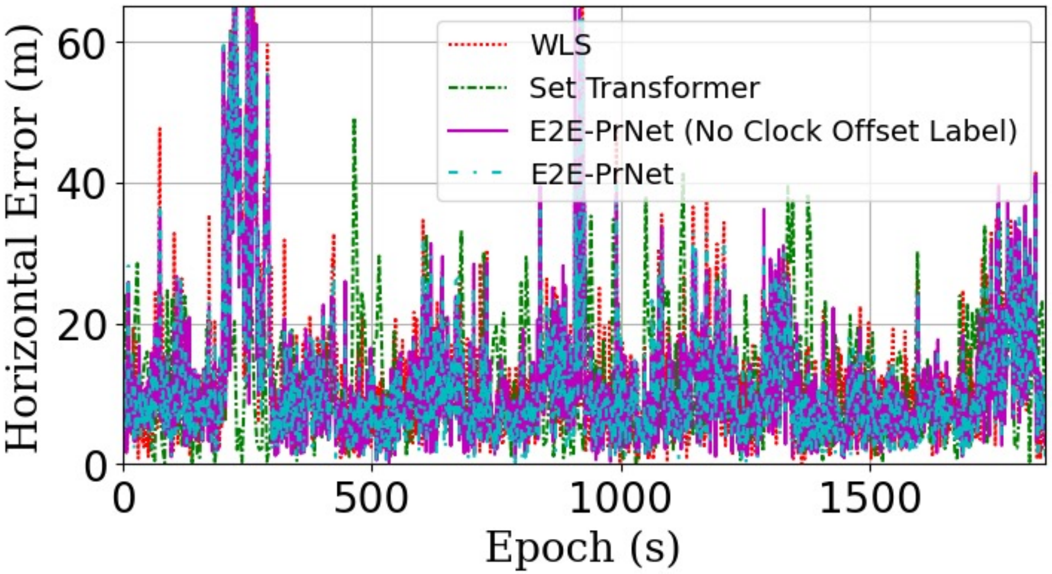
- The evaluation metrics adopted by Google:
  - ➢Calculate the horizontal distance errors between the predicted latitude/longitude and the ground truth latitude/longitude.
  - ➢Compute the mean of the 50^{th} and 95^{th} percentile horizontal distance errors.

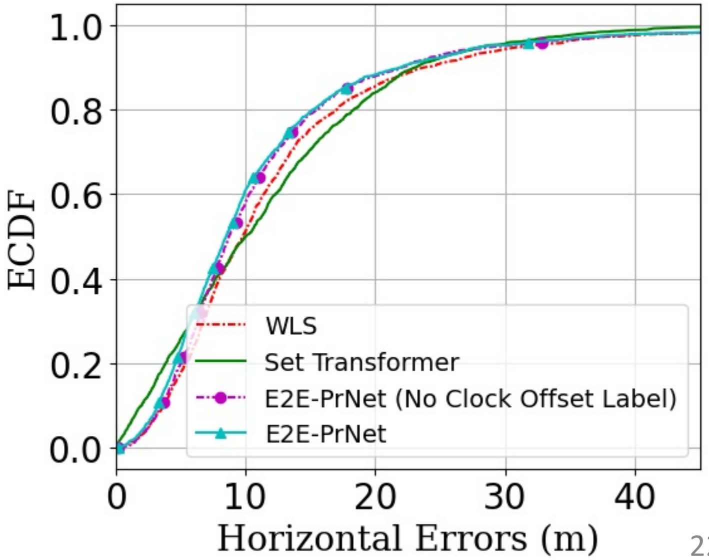
Experiments: Results (1/2)





Scenario I: rural fingerprinting





Scenario II: rural cross-trace

Experiments: Results (2/2)

Scenario I: rural fingerprinting

Scenario II: rural cross-trace

HORIZONTAL POSITIONING SCORES OF END-TO-END SOLUTIONS

Methods	Horizontal Score (meter)↓ Scenario I	Scenario II
WLS	16.390	20.666
Set Transformer	9.699 ($\mu = 15$m)	19.247 ($\mu = 22$m)
E2E-PrNet (No RCOL)	7.239	19.158
E2E-PrNet	**6.777**	**18.520**

RCOL: Receiver Clock Offset Label

Discussion: Does the Front-end PrNet Really Learn Pseudorange Errors? (1/2)

- We use the ground truth location and the WLS-based estimation of the user clock offset to train E2E-PrNet.

Substituting $x_k^* = x_{GT_k}$, $y_k^* = y_{GT_k}$, $z_k^* = z_{GT_k}$, and $\delta t_{u_k}^* = \delta \hat{t}_k^{(\mathrm{WLS})}$ into the pseudorange equation yields*:

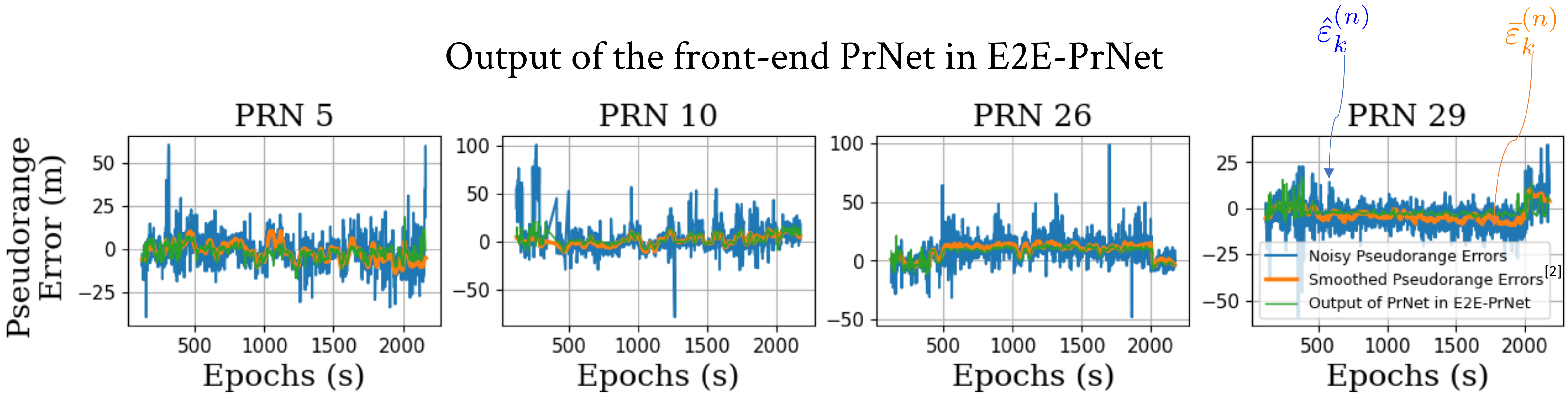
$$\hat{\varepsilon}_k^{(n)} = \underbrace{\bar{\varepsilon}_k^{(n)}}_{\text{Denoised error}} + \overbrace{e_k^{(n)}}^{\text{Unbiased noise}}$$

- The front-end PrNet is equivalently trained with ***noisy pseudorange errors***.

*Please refer to our paper for details

Discussion: Does the Front-end PrNet Really Learn Pseudorange Errors? (2/2)

- Neural network can learn information from noisy labels with regularization tricks, such as dropout and early stopping, when the signal-to-noise ratio is not too low[1]. *Neural network can filter noisy signals!*
- Thus, the output of the front-end PrNet should be $\hat{\varepsilon}_k^{(n)} \approx \bar{\varepsilon}_k^{(n)}$.



[1] M. Li, M. Soltanolkotabi, and S. Oymak, "Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks," in International conference on artificial intelligence and statistics. PMLR, 2020, pp. 4313–4324.

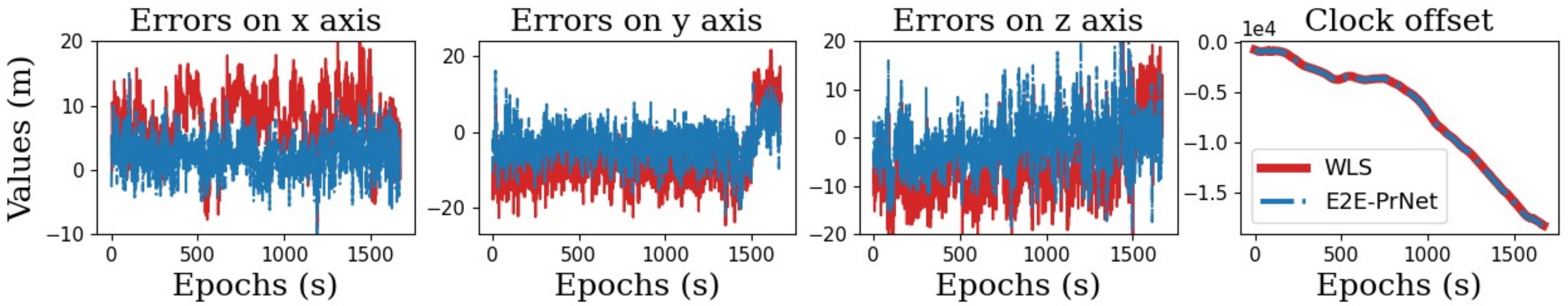
[2] X. Weng, K. Ling and H. Liu (2024), "PrNet: A Neural Network for Correcting Pseudoranges to Improve Positioning With Android Raw GNSS Measurements," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2024.3392302.

Discussion: How good results can E2E-PrNet deliver? (1/2)

- With the pseudorange error correction from the front-end PrNet, we get the state estimate of E2E-PrNet*:

$$\hat{x}_k = x_{GT_k} + \boxed{\mathbf{h}_{x_k}^T \mathbf{U}_k} \quad \hat{y}_k = y_{GT_k} + \boxed{\mathbf{h}_{y_k}^T \mathbf{U}_k} \quad \hat{z}_k = z_{GT_k} + \boxed{\mathbf{h}_{z_k}^T \mathbf{U}_k} \qquad \delta\hat{t}_{u_k} = \delta\hat{t}_{u_k}^{(\mathrm{WLS})}$$

Errors caused by unbiased noise



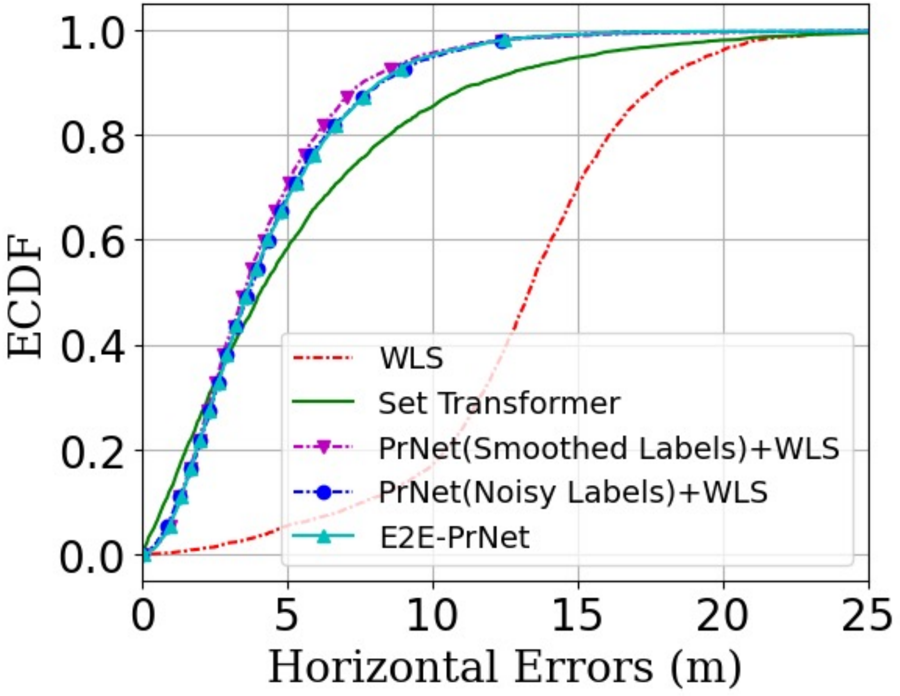
*Please refer to our paper for details

E2E-PrNet removes the biased positioning errors! 😉

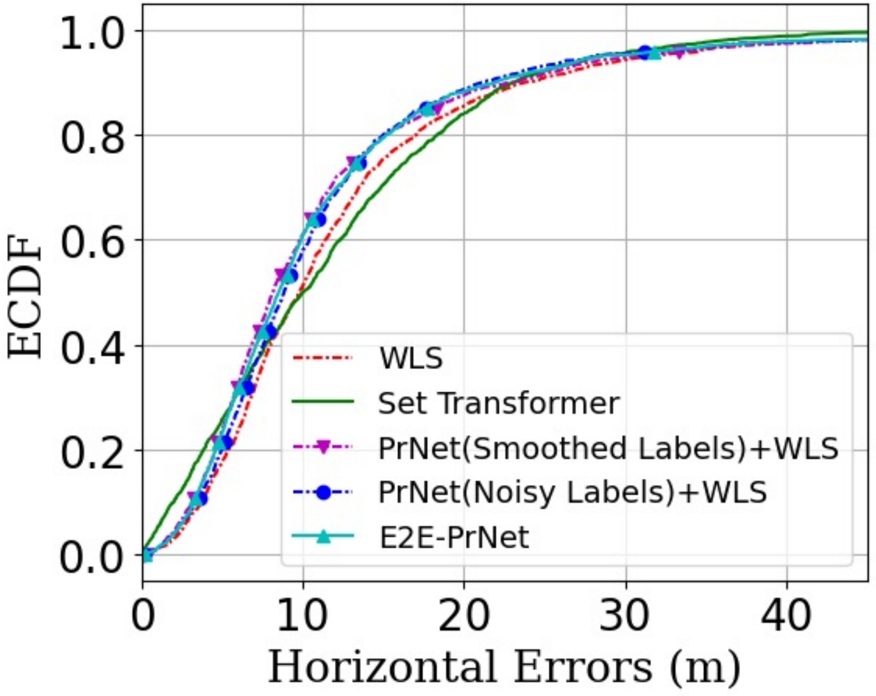
Discussion: How good results can E2E-PrNet deliver? (2/2)

- Compare E2E-PrNet and PrNet:

Scenario I



Scenario II



Methods	Horizontal Score (meter)↓	
	Scenario I	Scenario II
E2E-PrNet	6.777	18.520
PrNet+Noisy Labels	6.922	18.434
PrNet+Smoothed Labels	6.537	19.524

Recap

- Contribution:
  - We proposed an end-to-end learning framework, E2E-PrNet, to regress pseudorange errors by minimizing the final localization loss.
  - The key is ***the fusion of data-driven and model-based modules*** via a differentiable nonlinear least squares optimizer.

- Results:
  - E2E-PrNet has ***better*** performance than the classical WLS and the SOTA end-to-end data-driven method on the GSDC datasets.
  - E2E-PrNet is ***highly automatic*** and has an equivalent performance to PrNet trained with hand-crafted labels.

Thank you !

and

Q&A!