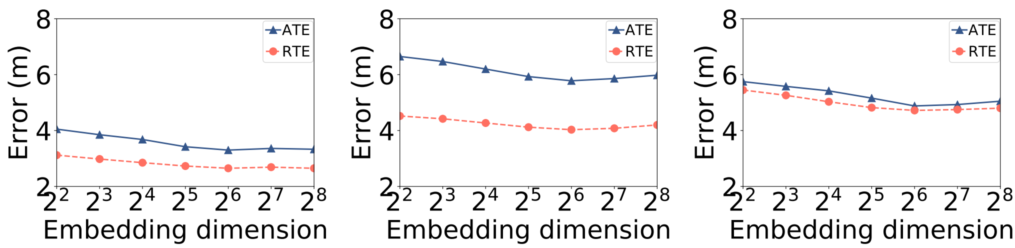
Appendix for Path Description Learning

1 MORE RESULTS ON SYSTEM COMPONENT ANALYSIS AND PATH RECOVERY

We further study the impact of embedding dimension and sequence length on our system performance here.

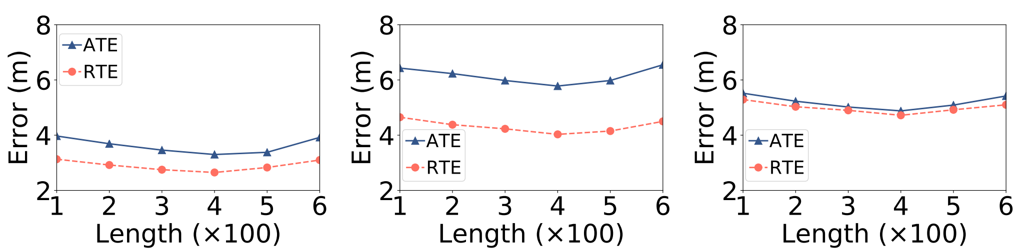
Embedding dimension: In SCAPE, we need to generate embeddings of certain dimension for feature learning of IMU signals. We show in Figure 1 the errors versus embedding dimension. The errors for the test seen data decrease and converge because higher embedding dimension learns better information from the data. In the test unseen and campus data, as the embedding dimension increases, the errors first decrease and then increase marginally, meaning that dimensions too high may overfit the data in training, hampering the generalization ability of the system.



(a) Test seen (b) Test unseen (c) Campus

Fig. 1. Errors versus Embedding Dimension.

Equally Divided Short Sequence length: In model training for SCAPE, we use short sequences with length of 400 IMU signal readings, which is equivalent to two-second interval. To study the impact of the sequence length, we show in Figure 2 errors versus sequence length. The errors first decrease and then increase because sequences too small may not contain enough information for a single pattern, such as swing, while sequences too large may have several walking contexts mixed up, making it difficult to extract useful features.



(a) Test seen (b) Test unseen (c) Campus

Fig. 2. Errors versus Sequence Length.

Extra path recovery results: We show extra illustrative results for path recovery in Figure 3, Figure 4 from the RoNIN dataset and Figure 5 from the campus dataset. Similar to what has been presented in the main text, SCAPE recovers paths best in both shape and orientation because signal patterns learned from short sequences are mostly from single context, and can be better used for velocity prediction. RoNIN [3] performs satisfactorily, but due to the potentially incorrect pattern matching for signals under different contexts, the results are not as promising. The CNN-based model [2] is not able to leverage fully the path information and cannot learn correlations well between signal patterns from the long sequence. A^3 [8] tries to find correct calibration opportunities for phone orientations, which may not be easy under unpredictable context changes, hampering the path recovery results. PDR [1] can hardly obtain users' walking direction under such drastic context changes, hence it does not demonstrate good performance in our scenario.

2 ONLINE TRAJECTORY COMPRESSION ALGORITHMS

We below explain the online trajectory compression algorithms that we have compared with our greedy algorithm. In online mode, GPS data points are continuously sampled, these algorithms decide which points to drop in real time.

- SQUISH[5]: A buffer size is pre-determined by the path length and compression ratio. It first accepts all the points when the buffer is not full. Each point (except for the first and the last one) in the buffer has a deviation value calculated from the path segment formed by its two consecutive neighboring points. When the buffer is full, upon the arrival of a new point, it first calculates the deviation value for the previously last point, and then removes the point in the buffer that has the smallest deviation value from the path. Finally, it adds the deviation value of the removed point to its two consecutive neighboring points.
- STTrace[6]: All the process is the same to SQUISH, except that after removing the point with the smallest deviation value, it recalculates the deviation value for each point (except for the first and the last one) in the buffer.
- OPW [4]: when an incoming point arrives, an anchor segment between the new point and the first point in the buffer is built to calculate the perpendicular Euclidean distance from every node in the buffer to the new segment. If the maximum distance among them is smaller than a threshold, the new point is abandoned and the algorithm proceeds. Otherwise, the new point is included into the buffer. Meanwhile, the corresponding point i with the maximum distance is accepted as a compressed trajectory point. The algorithm then iterates by setting i as the new starting point and repeat the process from point with index $i + 2$ again.
- Dead Reckoning[7]: It assumes that each object is moving in a constant velocity which can be derived from the recent historical data. Therefore it predicts the coordinates of each incoming point and calculates the Euclidean distance between coordinates of each incoming point and its prediction. If the distance is greater than a given threshold, it is kept as a delimiter. Otherwise, the new point is dropped.
- Uniform: It samples trajectory at a given rate and accepts the sampled points as compressed trajectory points.

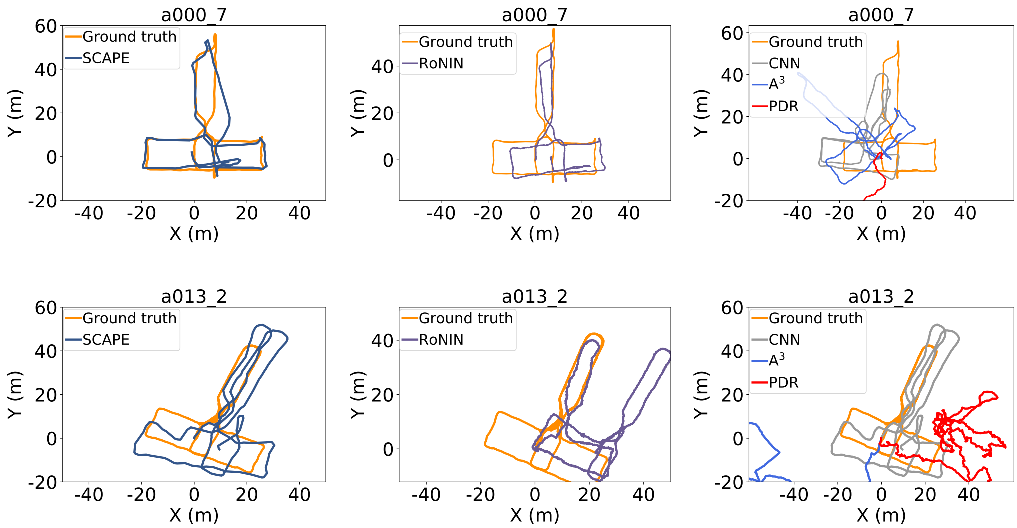


Fig. 3. Path recovery results (test seen).

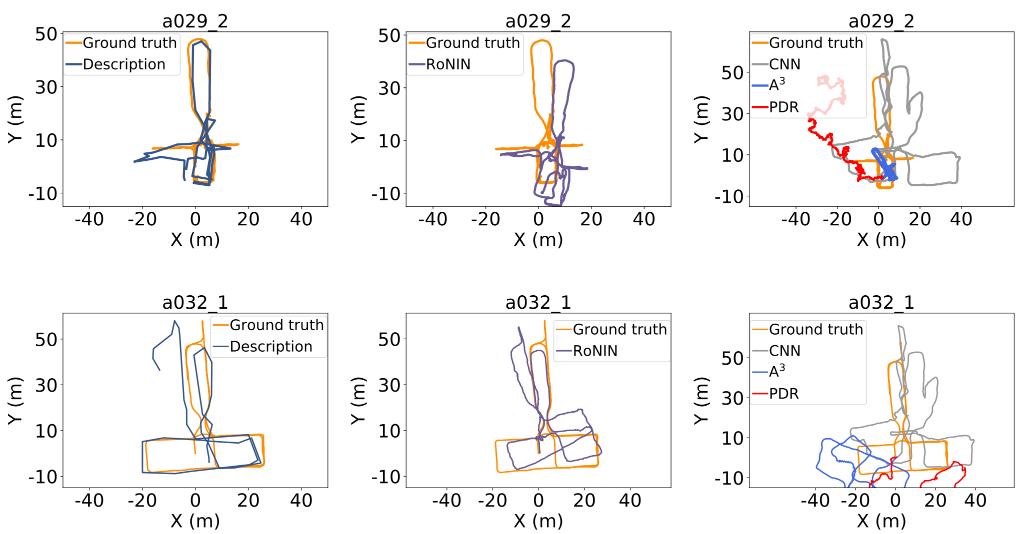


Fig. 4. Path recovery results (test unseen).

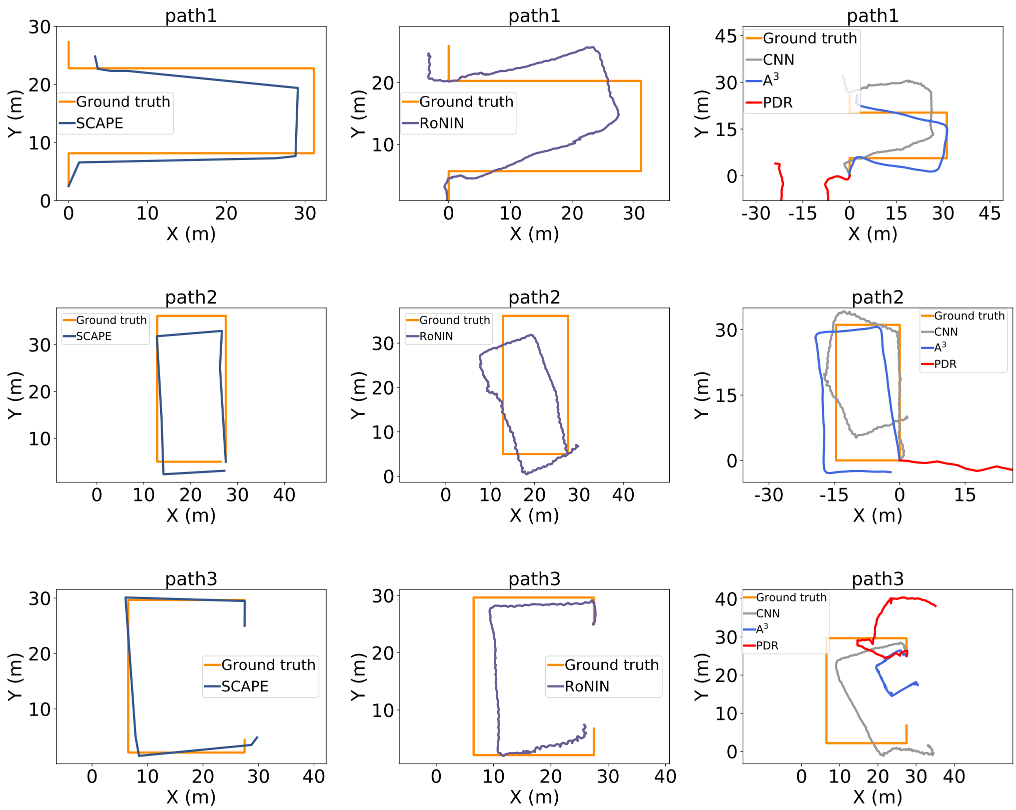


Fig. 5. Path recovery results (campus).

REFERENCES

[1] Agata Brajdic and Robert Harle. 2013. Walk Detection and Step Counting on Unconstrained Smartphones. In *ACM UbiComp*.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[3] S. Herath, H. Yan, and Y. Furukawa. 2020. RoNIN: Robust Neural Inertial Navigation in the Wild: Benchmark, Evaluations, & New Methods. In *IEEE ICRA*.

[4] Nirvana Meratnia et al. 2004. Spatiotemporal compression techniques for moving point objects. In *International Conference on Extending Database Technology*. Springer, 765–782.

[5] J. Muckell, J. Hwang, V. Patil, C. T Lawson, F. Ping, and SS Ravi. 2011. SQUISH: an online approach for GPS trajectory compression. In *COM. Geo.*

[6] Michalis Potamias, Kostas Patroumpas, and Timos Sellis. 2006. Sampling trajectory streams with spatiotemporal criteria. In *SSDBM'06*. IEEE.

[7] Goce Trajcevski, Hu Cao, Peter Scheuermanny, Ouri Wolfsonz, and Dennis Vaccaro. 2006. On-line data reduction and the quality of history in moving objects databases. In *MobiDE.*

[8] Pengfei Zhou, Mo Li, and Guobin Shen. 2014. Use It Free: Instantly Knowing Your Phone Attitude. In *ACM MobiCom.*