



SAPIENZA  
UNIVERSITÀ DI ROMA

LA SAPIENZA UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA, AUTOMATICA E  
GESTIONALE

---

# TRAJECTORY ENCODING

---

*Authors:*

Leandro de Souza Rosa

September 16, 2023

# 1 Trajectory Learning

## 2017 - Identifying Human Mobility via Trajectory Embeddings

[1] classifies users based on trajectory data. The problem is hard because there are many more trajectories than users.

[Recurrent Neural Network \(RNN\)](#) is used, and said to be good for classification when the number of labels is small. In particular uses a [Long Short-Term Memory \(LSTM\)](#) for processing sub-trajectories.

There is a location embedding, not sure how that is computed. But the trajectories are points on google maps, so there maybe be semantic information in there. The sequence of location embedding is passed onto the [LSTM](#), not sure how they handle different trajectory lengths.

## 2018 - Self-Consistent Trajectory Autoencoder: Hierarchical Reinforcement Learning with Trajectory Embeddings

Code on: <https://github.com/wyndwarrior/Sectar>

[2] learns an embedding for trajectories with one encoder and two decoders: a state decoder to decode from the latent space back into trajectories, and a policy decoder, which generates the trajectory in the environment. As such the state decoder predict the trajectory of the policy. The encoder is used in a hierarchical [Reinforcement Learning \(RL\)](#) setup.

The state encoder and decoder are [RNNs](#) and the policy decoder is a feed-forward [Neural Network \(NN\)](#).

**note:** if the encoder is trained with trajectories from different tasks, the policy will be conditioned to each task, what is sort of parameterizing the policy to tasks.

In the paper, the policy has unknown dynamics, and hence the [RL](#) setup. Trajectories are continuous poses of joints over time. Tested in simulation.

## 2018 - Anomalous Trajectory Detection Using Recurrent Neural Network

Code on: <https://github.com/LeeSongt/ATD-RNN>

[3] proposes anomalous trajectory detection using [RNN](#).

The trajectories are discretized, using a grid, and feed to a stacked [RNN](#) for learning the embedding, then a multi-layered perceptron and a soft-max layer detects if the trajectory is anomalous. The stacked [RNN](#) is made by feeding the hidden states of the previous to the next [RNN](#).

The trajectories are padded in order to get trajectories of the same length.

[LSTM](#) and [Gated Recurrent Unit \(GRU\)](#) are two special types of [RNNs](#) are tested. [GRUs](#) seems to work better.

## 2018 - Deep Representation Learning for Trajectory Similarity Computation

Code on: <https://github.com/boathit/t2vec>

[4] presents [t2vec](#). A [Deep Learning \(DL\)](#) approach for trajectory similarity. States that using [RNN](#) is not a very good idea because you cannot reconstruct the trajectory and it fails to consider spatial proximity, which is inherited in trajectory data.

called in the paper as [t2vec](#) or [seq2seq](#)?

The approach is based on the encoder-decoder framework. Handling varying sampling rates is done by augmenting the training data creating sub-trajectories by sub-sampling and noise addition. They also propose a spatial-aware loss, and pre-train the [cells](#) and let them to be optimized during training.

**Notes:** The paper is very confusing. I do not really know that are the inputs and outputs or how the sequences are fed in the [RNN](#) inside the encoder.

## 2019 - Computing Trajectory Similarity in Linear Time: A Generic Seed-Guided Neural Metric Learning Approach

[5] proposes a method for accelerating trajectory similarity computation by sampling seeds of trajectories, computing their similarity, and approximating them with a neural metric.

States that [RNNs](#), [LSTMs](#), and [GRUs](#) can only model one sequence without considering the between-sequence correlation.

Does not consider time in the trajectory. Starts sampling from the trajectories and computes a distance matrix between the samples using a given trajectory distance metric which is then normalized.

The **RNN** is augmented with a memory, which is created by dividing the space into a grid, and for each grid slot, the memory stores the hidden vector of the **RNN**. This memory is used to extend the **RNN** cell, sort of like an **LSTM**.

The loss for training is  $\mathcal{L}_{\tau_i, \tau_j} = \sum_k w_k (f(\tau_i, \tau_j) - \exp(-||e_i - e_j||))$ , a weighted difference between the similarity metric  $f$  and the distance in the embedding space  $(e_i - e_j)$ . The weight  $w_k$  is obtained using the normalized distance matrix, computing pairs of similar and dissimilar trajectories and more fancy stuff.

**Notes:** map is like google map.

## **2020 - Trembr: Exploring Road Networks for Trajectory Representation Learning**

[6] presents **traj2vec**. The paper focus on trajectories on roads. It preprocesses the trajectories by projecting them in a road network and the trajectory is a sequence of road segments and travel time.

The **RNN** decoder is conditioned to the road network, and the training is made by optimising a loss for the trajectory and another for time.

**Notes:** Maybe the secret for velocities profiles is in the addition of time to the loss.

## **2020 - Trajectory similarity learning with auxiliary supervision and optimal matching**

[7]

## **2020 - Semantic trajectory representation and retrieval via hierarchical embedding**

[8]

## **2020 - MARC: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings**

[9]

## **2021 - Multiple Object Tracking by Trajectory Map Regression with Temporal Priors Embedding**

[10]

## **2021 - Embedding-Based Similarity Computation for Massive Vehicle Trajectory Data**

[11]

## **2021 - STENet: A hybrid spatio-temporal embedding network for human trajectory forecasting**

[12]

## **2021 - A Graph-Based Approach for Trajectory Similarity Computation in Spatial Networks**

[13]

## **2021 - T3S: Effective Representation Learning for Trajectory Similarity Computation**

[14]

## **2021 - How meaningful are similarities in deep trajectory representations?**

[15]

**2022 - Deep Fuzzy Contrast-Set Deviation Point Representation and Trajectory Detection**

[\[16\]](#)

**2022 - Contrastive Pre-training of Spatial-Temporal Trajectory Embeddings**

[\[17\]](#)

**2022 - TMN: Trajectory Matching Networks for Predicting Similarity**

[\[18\]](#)

**2022 - TSNE: Trajectory Similarity Network Embedding**

[\[19\]](#)

**2022 - Towards robust trajectory similarity computation: Representation-based spatio-temporal similarity quantification**

[\[20\]](#)

**2023 - GRLSTM: Trajectory Similarity Computation with Graph-Based Residual LSTM**

[\[21\]](#)

**2023 - Spatial-temporal fusion graph framework for trajectory similarity computation**

[\[22\]](#)

**2023 - Contrastive Trajectory Similarity Learning with Dual-Feature Attention**

[\[23\]](#)

**2023 - Spatio-Temporal Trajectory Similarity Measures: A Comprehensive Survey and Quantitative Study**

[\[24\]](#)

## **2 Trajectory Learning on Robotics**

**1991 - Solving the moving obstacle path planning problem using embedded variational methods**

[\[25\]](#)

**2016 - Robobarista: Learning to Manipulate Novel Objects via Deep Multimodal Embedding**

[\[26\]](#)

**2020 - Controlling Assistive Robots with Learned Latent Actions**

[\[27\]](#)

**2020 - DiversityGAN: Diversity-Aware Vehicle Motion Prediction via Latent Semantic Sampling**

[\[28\]](#)

**2021 - Use of Embedding Spaces for Transferring Robot Skills in Diverse Sensor Settings**

[\[29\]](#)

**2022 - STL2vec: Signal Temporal Logic Embeddings for Control Synthesis With Recurrent Neural Networks**

[\[30\]](#)

**2022 - Learning latent actions to control assistive robots**

[\[31\]](#)

**2022 - Promoting Quality and Diversity in Population-based Reinforcement Learning via Hierarchical Trajectory Space Exploration**

[\[32\]](#)

**2023 - SIRL: Similarity-Based Implicit Representation Learning**

[\[33\]](#)

### **3 Comparison**

### **4 Challenges**



## References

- [1] Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. Identifying Human Mobility via Trajectory Embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 1689–1695. AAAI Press, 2017.
- [2] John Co-Reyes, YuXuan Liu, Abhishek Gupta, Benjamin Eysenbach, Pieter Abbeel, and Sergey Levine. Self-Consistent Trajectory Autoencoder: Hierarchical Reinforcement Learning with Trajectory Embeddings. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1009–1018. PMLR, 10–15 Jul 2018.
- [3] Li Song, Ruijia Wang, Ding Xiao, Xiaotian Han, Yanan Cai, and Chuan Shi. Anomalous Trajectory Detection Using Recurrent Neural Network. In Guojun Gan, Bohan Li, Xue Li, and Shuliang Wang, editors, *Advanced Data Mining and Applications*, pages 263–277, Cham, 2018. Springer International Publishing.
- [4] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S. Jensen, and Wei Wei. Deep Representation Learning for Trajectory Similarity Computation. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 617–628, 2018.
- [5] Di Yao, Gao Cong, Chao Zhang, and Jingping Bi. Computing Trajectory Similarity in Linear Time: A Generic Seed-Guided Neural Metric Learning Approach. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1358–1369, 2019.
- [6] Tao-Yang Fu and Wang-Chien Lee. Trembr: Exploring Road Networks for Trajectory Representation Learning. *ACM Trans. Intell. Syst. Technol.*, 11(1), feb 2020.
- [7] Xingyu; JIANG Qize; ZHENG Baihua; SUN Zhenbang; SUN Weiwei; ZHANG, Hanyuan; ZHANG and Changhu WANG. Trajectory similarity learning with auxiliary supervision and optimal matching. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*,, pages 3209–3215. Research Collection School Of Information Systems, 11–17 Jul 2020.
- [8] Chongming Gao, Zhong Zhang, Chen Huang, Hongzhi Yin, Qinli Yang, and Junming Shao. Semantic trajectory representation and retrieval via hierarchical embedding. *Information Sciences*, 538:176–192, 2020.
- [9] Lucas May Petry, Camila Leite Da Silva, Andrea Esuli, Chiara Renso, and Vania Bogorny. MARC: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings. *International Journal of Geographical Information Science*, 34(7):1428–1450, 2020.
- [10] Xingyu Wan, Sanping Zhou, Jinjun Wang, and Rongye Meng. Multiple Object Tracking by Trajectory Map Regression with Temporal Priors Embedding. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 1377–1386, New York, NY, USA, 2021. Association for Computing Machinery.
- [11] Yuanyi Chen, Peng Yu, Wenwang Chen, Zengwei Zheng, and Minyi Guo. Embedding-Based Similarity Computation for Massive Vehicle Trajectory Data. *IEEE Internet of Things Journal*, 9(6):4650–4660, 2022.
- [12] Bo Zhang, Chengzhi Yuan, Tao Wang, and Hongbo Liu. STENet: A hybrid spatio-temporal embedding network for human trajectory forecasting. *Engineering Applications of Artificial Intelligence*, 106:104487, 2021.
- [13] Peng Han, Jin Wang, Di Yao, Shuo Shang, and Xiangliang Zhang. A Graph-Based Approach for Trajectory Similarity Computation in Spatial Networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 556–564, New York, NY, USA, 2021. Association for Computing Machinery.
- [14] Peilun Yang, Hanchen Wang, Ying Zhang, Lu Qin, Wenjie Zhang, and Xuemin Lin. T3S: Effective Representation Learning for Trajectory Similarity Computation. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2183–2188, 2021.
- [15] Saeed Taghizadeh, Abel Elekes, Martin Schäler, and Klemens Böhm. How meaningful are similarities in deep trajectory representations? *Information Systems*, 98:101452, 2021.

- [16] Usman Ahmed, Jerry Chun-Wei Lin, and Gautam Srivastava. Deep Fuzzy Contrast-Set Deviation Point Representation and Trajectory Detection. *IEEE Transactions on Fuzzy Systems*, 31(2):571–581, 2023.
- [17] Yan Lin, Huaiyu Wan, Shengnan Guo, and Youfang Lin. Contrastive Pre-training of Spatial-Temporal Trajectory Embeddings, 2022.
- [18] Peilun Yang, Hanchen Wang, Defu Lian, Ying Zhang, Lu Qin, and Wenjie Zhang. TMN: Trajectory Matching Networks for Predicting Similarity. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1700–1713, 2022.
- [19] Jiaxin Ding, Bowen Zhang, Xinbing Wang, and Chenghu Zhou. TSNE: Trajectory Similarity Network Embedding. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems, SIGSPATIAL '22*, New York, NY, USA, 2022. Association for Computing Machinery.
- [20] Ziwen Chen, Ke Li, Silin Zhou, Lisi Chen, and Shuo Shang. Towards robust trajectory similarity computation: Representation-based spatio-temporal similarity quantification. *World Wide Web*, pages 1–24, 2022.
- [21] Silin Zhou, Jing Li, Hao Wang, Shuo Shang, and Peng Han. GRLSTM: Trajectory Similarity Computation with Graph-Based Residual LSTM. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4):4972–4980, Jun. 2023.
- [22] Silin Zhou, Peng Han, Di Yao, Lisi Chen, and Xiangliang Zhang. Spatial-temporal fusion graph framework for trajectory similarity computation. *World Wide Web*, 26(4):1501–1523, 2023.
- [23] Yanchuan Chang, Jianzhong Qi, Yuxuan Liang, and Egemen Tanin. Contrastive Trajectory Similarity Learning with Dual-Feature Attention. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2933–2945, 2023.
- [24] Danlei Hu, Lu Chen, Hanxi Fang, Ziquan Fang, Tianyi Li, and Yunjun Gao. Spatio-Temporal Trajectory Similarity Measures: A Comprehensive Survey and Quantitative Study, 2023.
- [25] H. Tominaga and B. Bavarian. Solving the moving obstacle path planning problem using embedded variational methods. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 450,451,452,453,454,455, Los Alamitos, CA, USA, apr 1991. IEEE Computer Society.
- [26] Jaeyong Sung, Seok Hyun Jin, Ian Lenz, and Ashutosh Saxena. Robobarista: Learning to Manipulate Novel Objects via Deep Multimodal Embedding, 2016.
- [27] Dylan P. Losey, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, and Dorsa Sadigh. Controlling Assistive Robots with Learned Latent Actions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–384, 2020.
- [28] Xin Huang, Stephen G. McGill, Jonathan A. DeCastro, Luke Fletcher, John J. Leonard, Brian C. Williams, and Guy Rosman. DiversityGAN: Diversity-Aware Vehicle Motion Prediction via Latent Semantic Sampling. *IEEE Robotics and Automation Letters*, 5(4):5089–5096, 2020.
- [29] Kazushi Ninomiya, Masakazu Hirokawa, and Kenji Suzuki. Use of Embedding Spaces for Transferring Robot Skills in Diverse Sensor Settings. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 547–551, 2021.
- [30] Wataru Hashimoto, Kazumune Hashimoto, and Shigemasa Takai. STL2vec: Signal Temporal Logic Embeddings for Control Synthesis With Recurrent Neural Networks. *IEEE Robotics and Automation Letters*, 7(2):5246–5253, 2022.
- [31] Dylan P Losey, Hong Jun Jeon, Mengxi Li, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, Jeannette Bohg, and Dorsa Sadigh. Learning latent actions to control assistive robots. *Autonomous robots*, 46(1):115–147, 2022.
- [32] Jiayu Miao, Tianze Zhou, Kun Shao, Ming Zhou, Weinan Zhang, Jianye Hao, Yong Yu, and Jun Wang. Promoting Quality and Diversity in Population-based Reinforcement Learning via Hierarchical Trajectory Space Exploration. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7544–7550, 2022.



- [33] Andreea Bobu, Yi Liu, Rohin Shah, Daniel S. Brown, and Anca D. Dragan. SIRL: Similarity-Based Implicit Representation Learning. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '23, page 565–574, New York, NY, USA, 2023. Association for Computing Machinery.