Trajectory-Embedded Matryoshka Representation Learning for Enhanced Similarity Analysis

Federico Pennino[1], Andrea Gurioli[1] and Maurizio Gabbrielli[1] *

1 - University of Bologna - DISI
Via Zamboni, 33, 40126 Bologna - Italy

Abstract. This paper introduces Trajectory-Embedded Matryoshka Representation Learning (TE-MRL). This novel framework synergies the capabilities of trajectory representation learning with the adaptability and efficiency of Matryoshka Representation Learning (MRL). TE-MRL is engineered to generate adaptive, multi-granular embeddings that efficiently capture the spatial-temporal dynamics inherent in trajectory data. We evaluate TE-MRL on the Porto dataset, focusing on trajectory similarity and k-nearest trajectory similarity tasks. Our findings demonstrate that TE-MRL preserves critical features such as travel semantics and temporal regularities while it can significantly reduce computational time and memory footprint. The proposed approach matches existing methods' accuracy and efficiency but demonstrates robust adaptability under varying computational constraints. Furthermore, we proposed a two-stage retrieval pipeline to enhance computational time while maintaining the same precision. We reduced the computation time by 8× while maintaining state-of-the-art precision. The effectiveness of TE-MRL in handling the complexity of the Porto dataset underlines its potential for broader applications in urban computing and mobility analytics.

1 Introduction

Representation of spatial-temporal data, known as Trajectory Representation Learning (TRL), has gained prominence with the increasing demand for machine learning approaches that extract meaningful spatial features, enhancing data analysis in GPS-based systems [1]. Tasks such as trajectory search and trajectory similarity are just a few examples of the trajectory representation scenario, where we can highlight two main weaknesses: (1) retrieval efficiency (time required to search for the most similar trajectory in the database), which becomes a compromising factor [2]; (2) exhaustion of memory in low resource devices [3].

Deep metric learning systems became a standard in these tasks [1, 2, 4], dropping the initial quadratic time complexity to a linear complexity [2]. While these techniques aim to balance the inherent trade-off between efficiency and effectiveness, the time required for this lookup algorithm remains closely tied to the embeddings' dimensionality and associated memory usage. This leads to choosing between higher-dimensional vectors with strong representation capabilities and less informative lower-dimensional vectors.

*This work was supported by the NextGenerationEU fund.

In order to address this trade-off, Kusupati et al. [5] introduced *Matryoshka Representation Learning* (MRL). This framework uses a training loss to output a vector that maintains good representation capabilities across various dimensional granularities. Starting from [1], this paper aims to enhance efficiency in trajectory retrieval over time and memory usage by leveraging the training loss presented by Kusupati et al. [5]. We propose a reproducible pipeline that uses the Matryoshka approach and is adaptable to various Trajectory Representation Learning tasks. In particular, the main contributions of this paper are the following:

- A graph neural network able to nest information at various levels of granularity within a single vector, enhancing model adaptability and efficiency without increasing computational overhead.
- A retrieval pipeline, called *two-step search approach*, capable of increasing the efficiency by a factor of $8\times$ while maintaining high accuracies over the two analyzed different tasks.

By incorporating the Matryoshka loss [5] we obtain a refinement of the architecture proposed in [1] which produces smaller embeddings without sacrificing performance. This allows us to enhance retrieval efficiency while reducing both computational and storage demands.

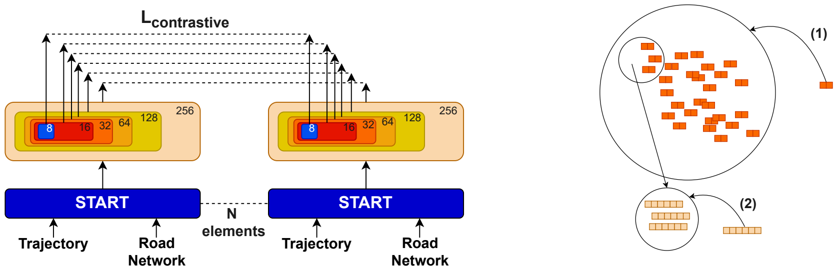
2 Related work

Several studies [6, 7] have investigated trajectory representation using temporal-spatial features, producing embeddings for tasks like trajectory similarity search. Deep learning models, particularly sequence-to-sequence (seq-to-seq) architectures such as LSTMs, have achieved state-of-the-art performance in this domain.

Li et al.[6] identified a linear correlation between dataset size and retrieval time, emphasizing the challenges of scaling. Traditional methods often relied on a "one model, one task" paradigm, addressing tasks like clustering or travel time prediction in isolation. In contrast, Jiang et al.[1] introduced a versatile trajectory representation model with robust feature extraction capabilities, achieving strong performance across multiple tasks on the Porto dataset: $1.897 \cdot 10^{-3}$ MR for similarity, $0.890 \cdot 10^{-4}$ accuracy for classification, and $1.334 \cdot 10^{-3}$ MAE for travel time estimation. As mentioned in the previous section, our work builds on the findings of Jiang et al. findings [1] and of Kusupati et al. [5].

3 Methodology

We adapted the methodology of Jiang et al. [1], embedding spatial and temporal information with a modified loss function using the Matryoshka technique to optimize multiple output dimensions. The model performance was evaluated in relation to embedding dimensionalities and memory usage.



(a) $\mathcal{L}^{con}$ Contrastive loss

(b) Two-stage retrieval

Fig. 1: Main modification to Jiang et al. [1] work. (a) shows $\mathcal{L}^{con}$, the Matryoshka contrastive loss. (b) presents the two-stage retrieval process.

3.1 Spatial-Temporal architecture

Jiang et al. [1] introduced *"START"*, an architecture to encode temporal (time of the day) and spatial information (trajectory information and road network) jointly. The architecture encodes trajectory information by adding time-of-day and day-of-week information to the representation. The architecture uses a graph neural network to convert the road network into road representation vectors. To support multiple downstream tasks, the work implemented two different losses:

- $\mathcal{L}^{mask}$: *Span-Masked Trajectory Recovery*, where part of the input road network is masked, the model tries to recover it. This loss captures co-occurrence relationships between roads and contextual information of the road network.
- $\mathcal{L}^{con}$: *Trajectory Contrastive Learning*, where similar vectorial representations are brought closer in latent space. This objective is meant to improve the modeling of the spatial-temporal characteristics and travel semantics.

The final pre-training loss $\mathcal{L}^{pre}$ is defined by the formula in Eq. 1:

$$\mathcal{L}^{pre} = (1-\lambda)\,\mathcal{L}^{mask} + \lambda\,\mathcal{L}^{con} \tag{1}$$

where λ acts as a weight, balancing the two tasks.

3.2 Efficiency analysis with Matryoshka loss

We adapted the former architecture to obtain a model capable of producing a vector that maintains good semantic representation capabilities even when truncated by the last dimensions. In these terms, we modified the *Trajectory Contrastive Learning* objective by incorporating Matryoshka loss [5].

Model	Most Similar Search					Trajectory Search
	MR	**MRR**	**HRQ1**	**HRQ5**	**HR@10**	**Precision**
No Matryoshka	**1.3**	0.98	0.97	**0.99**	**0.99**	**0.80**
256	**1.3**	**0.99**	**0.98**	**0.99**	**0.99**	0.79
128	1.4	**0.99**	**0.98**	**0.99**	**0.99**	0.78
64	1.5	0.99	0.98	0.995	0.997	0.75
32	2.0	0.98	0.97	0.99	0.994	0.70
16	12	0.95	0.94	0.97	0.98	0.61
8	31	0.85	0.80	0.90	0.93	0.50

Table 1: Performance comparison of model metrics with and without Matryoshka Loss across various embedding sizes in trajectory analysis. The tasks are Most Similar Search and 5-Nearest Trajectory Search.

This is illustrated in Fig. 1a where the $\mathcal{L}^{con}$ in-batch contrastive loss used by Jiang et al. [1] aligns similar trajectories with the same dimensionality while moving away from the other representations—also with the same dimensionality—present in the batch. We have composed each batch of $N = 32$ elements, each containing a couple of similar trajectories treated as ground truth, where each trajectory is represented with the trajectory and the road network representation (see Section 3.1). We applied the Matryoshka loss [5] to a set d of different dimensionalities where $d = \{256, 128, 32, 16, 8\}$, obtaining $|d|$ different losses, subsequently summated in $\sum_{dim \in d} \mathcal{L}^{con}_{dim}$. We have that the final loss is represented by Eq. 2:

$$\mathcal{L}^{pre} = (1 - \lambda)\, \mathcal{L}^{mask} + \lambda \sum_{dim \in d} \mathcal{L}^{con}_{dim} \tag{2}$$

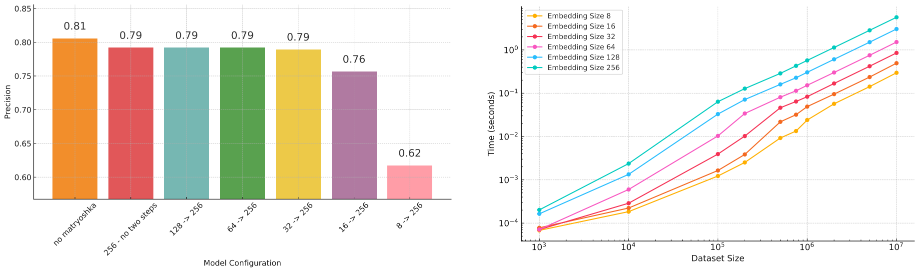
The parameter λ balances the importance of the masked contrastive loss, $\mathcal{L}^{mask}$, against the summation of contrastive losses across different embedding dimensions, $\mathcal{L}^{con}_{d}$, ensuring that the network does not overfit a specific task.

We trained the model with AdamW as an optimizer, with a batch size of 32 elements for 20 training epochs. We initially set the learning rate at $2 \cdot 10^{-4}$ with a warmup for the first four epochs and decreases using a cosine annealing schedule. The λ weight is settled at 0.6. We evaluated the model with different dimensionalities on the Trajectory Similarity Search task and k-nearest trajectory search with the Euclidean distance. We assessed the efficiency by relating the model's performance to time and memory consumption.

3.3 Two-stage retrieval pipeline

We designed a two-step retrieval (Figure 1b) to optimize the time search while maintaining the same model performance. We define the stages as follows: (1) First lookup phase: we use a lower dimensionality to shrink the search space to a small subset of 1000 samples. (2) Second lookup phase: we use the full dimensionality to retrieve the most similar sample out of the 1000 subset.

We conducted a grid search over the vectors' dimensionality and the subset's dimension to obtain the best-performing algorithm.



(a) Precision Two-Step Approach (b) Execution time over dataset sizes

Fig. 2: Performance analysis of embedding sizes using matryoshka Loss. (a) shows the precision achieved at various embedding sizes with Two-Step. (b) charts the computational time required at different dataset sizes for varying embedding sizes.

4 Discussion

The results presented in this study underscore the efficacy of Trajectory-Embedded Matryoshka Representation Learning (TE-MRL) in managing the demands of computational resources in trajectory data analysis. By incorporating MRL into trajectory analysis, TE-MRL facilitates a hierarchical yet computationally efficient approach to understanding spatial-temporal dynamics, as evidenced by the results on the Porto dataset. As shown in Table 1, TE-MRL maintains high levels of accuracy in Most Similar Search and Trajectory Search tasks across various embedding sizes. Even at reduced dimensions (e.g., 16 and 32), TE-MRL achieves good results in terms of Mean Reciprocal Rank (MRR) and Hit Rates (HR). This adaptability is crucial for applications where computational resources are limited or where real-time data processing is required.

Fig. 2a shows the performance of the two-stage search approach within the TE-MRL framework, focusing mainly on the precision achieved with initial embeddings of varying dimensions transitioning to a 256-dimensional space.

We showed that by employing 32 dimensions during the initial retrieval stage and subsequently transitioning to 256 dimensions for the second stage, we matched the performance of the 256-dimensional matryoshka single-stage approach with a 8× speed-up in time. This precision result underscores the efficacy of the TE-MRL framework in utilizing lower-dimensional embeddings to achieve computational efficiency without compromising the quality of the results.

Fig. 2b demonstrates the scalability of TE-MRL across different dataset sizes and embedding dimensions. The ratio of computational time related to the size of the dataset decreases by half with each dimensionality reduction. Hence, the two-step retrieval process helps to moderate the time complexity of computation by using smaller embeddings in the initial retrieval phase while ensuring that precision is not compromised.

5 Conclusion

In this work, we present a promising approach to trajectory data analysis that combines the strengths of Graph Neural Network with Matryoshka Representation Learning to offer a scalable and efficient solution. Its application could extend beyond the tested scenarios, providing valuable insights into various real-world applications that require efficient data processing and retrieval capabilities. The TE-MRL framework leverages MRL to encapsulate information at multiple granularities within a single vector, significantly reducing memory usage while ensuring seamless adaptability to various computational constraints. Our two-stage retrieval pipeline achieves precision levels close to state-of-the-art methods - within a -2% gap -, delivering an 8 times improvement in computational efficiency. By accepting a -5% trade-off in precision, the speedup increases to 16 times. This approach effectively balances latency and memory requirements while maintaining high performance and accuracy across a wide range of applications.

References

[1] Jiawei Jiang, Dayan Pan, Houxing Ren, Xiaohan Jiang, Chao Li, and Jingyuan Wang. Self-supervised trajectory representation learning with temporal regularities and travel semantics. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*, pages 843–855. IEEE, 2023.

[2] Di Yao, Gao Cong, Chao Zhang, and Jingping Bi. Computing trajectory similarity in linear time: A generic seed-guided neural metric learning approach. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*, pages 1358–1369. IEEE, 2019.

[3] Jiajun Liu, Kun Zhao, Philipp Sommer, Shuo Shang, Brano Kusy, Jae-Gil Lee, and Raja Jurdak. A novel framework for online amnesic trajectory compression in resource-constrained environments. *IEEE Trans. Knowl. Data Eng.*, 28(11):2827–2841, 2016.

[4] Yile Chen, Xiucheng Li, Gao Cong, Zhifeng Bao, Cheng Long, Yiding Liu, Arun Kumar Chandran, and Richard Ellison. Robust road network representation learning: When traffic patterns meet traveling semantics. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 211–220. ACM, 2021.

[5] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham M. Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9*, 2022.

[6] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S. Jensen, and Wei Wei. Deep representation learning for trajectory similarity computation. In *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*, pages 617–628. IEEE Computer Society, 2018.

[7] Tao-Yang Fu and Wang-Chien Lee. Trembr: Exploring road networks for trajectory representation learning. *ACM Trans. Intell. Syst. Technol.*, 11(1):10:1–10:25, 2020.