2024 IEEE 27th International Conference on
Intelligent Transportation Systems (ITSC)
September 24- 27, 2024. Edmonton, Canada

Pedestrian Motion Prediction Using Transformer-based Behavior Clustering and Data-Driven Reachability Analysis

Kleio Fragkedaki, Frank J. Jiang, Karl H. Johansson, Jonas Mårtensson

***Abstract*— In this work, we present a transformer-based framework for predicting future pedestrian states based on clustered historical trajectory data. In previous studies, researchers propose enhancing pedestrian trajectory predictions by using manually crafted labels to categorize pedestrian behaviors and intentions. However, these approaches often only capture a limited range of pedestrian behaviors and introduce human bias into the predictions. To alleviate the dependency on manually crafted labels, we utilize a transformer encoder coupled with hierarchical density-based clustering to automatically identify diverse behavior patterns, and use these clusters in data-driven reachability analysis. By using a transformer-based approach, we seek to enhance the representation of pedestrian trajectories and uncover characteristics or features that are subsequently used to group trajectories into different "behavior" clusters. We show that these behavior clusters can be used with data-driven reachability analysis, yielding an end-to-end data-driven approach to predicting the future motion of pedestrians. We train and evaluate our approach on a real pedestrian dataset, showcasing its effectiveness in forecasting pedestrian movements.**

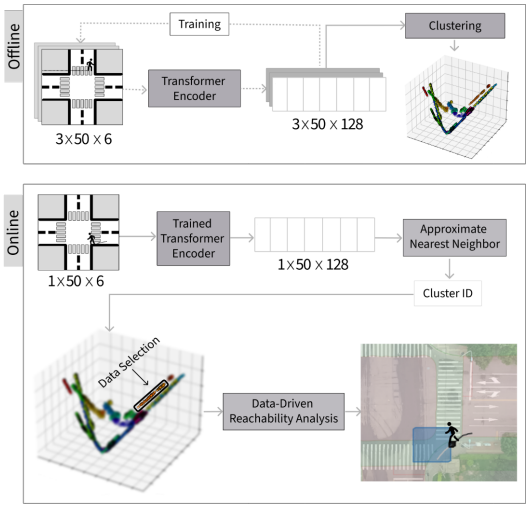


Fig. 1: Overview of the proposed framework

I. Introduction

Today, pedestrians continue to pose a challenge for ensuring the safety of intelligent transport systems. Pedestrians are able to freely move around in the same space as vehicles and often cross paths with vehicles. In such scenarios, the vehicles need to ensure the safety of the pedestrians, regardless of the pedestrians' actions. This poses a significant challenge for automated vehicles since pedestrians are both difficult to model and have a large degree of freedom. Due to this challenge, ensuring their safety often results in overly conservative driving policies.

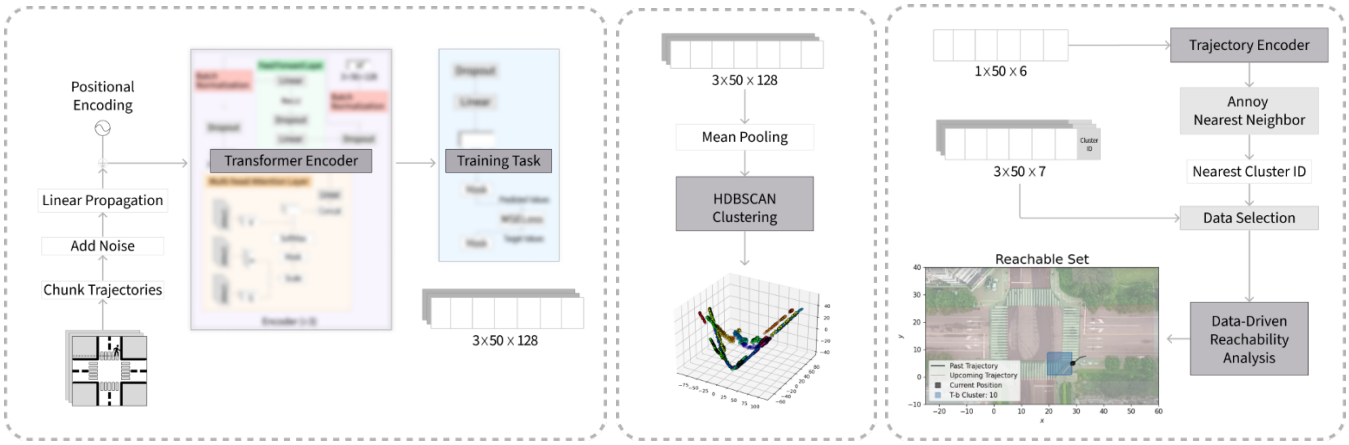
A popular approach for enhancing the safety of intelligent transport systems is the integration of reachability analysis [1]–[3]. By using reachability analysis, we are able to predict an over-approximation of the set of all possible places a pedestrian can be in the future, regardless of what specific decisions they make. Then, we can incorporate these sets into the path planning and control system of a vehicle, so it can make safe decisions around pedestrians. However, when reachability analysis is applied to predict the future motion of pedestrians, two issues arise: (1) there is little consensus on accurate models for pedestrian motion and (2) even if a model is chosen, the resultant reachable sets end up being very conservative due to the pedestrians' decision freedom. To address the first issue, several have proposed the use of data-driven reachability analysis for agents that are difficult to model [4], [5]. The use of data-driven reachability analysis removes the dependency on choosing an accurate model for pedestrians, however, data-driven reachability analysis can also suffer from being overly conservative [5]. To address the overly conservative reachable sets, authors in [3], [6] explore the use of both assumed side information and behavior modes to reduce the conservativeness of the predicted reachable sets. Although there is indication that this approach can yield reachable sets that are less conservative while still providing safety benefits, a key challenge with this approach is the prediction of the behavior mode.

To address this challenge, current approaches use manually engineered and labeled systems for pedestrian prediction, which often rely on fixed and pre-defined labels [3], [7], [8]. However, this manual labeling process can introduce bias and often fails to encompass the diverse range of pedestrian behaviors observed across various contexts. Alternatively, there are approaches that use clustering to identify movement patterns through various trajectory clustering techniques [9], [10]. Inspired by these approaches, we are interested in exploring the use of transformers to identify clusters, since transformers have demonstrated potential in capturing com-

This work was partially supported by the Swedish Innovation agency (Vinnova), under grant 2021-02555 Future 5G Ride, within the Strategic Vehicle Research and Innovation program (FFI); the Wallenberg Artificial Intelligence, Autonomous Systems, and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation; the Swedish Research Council Distinguished Professor Grant 2017-01078; and the Knut and Alice Wallenberg Foundation Wallenberg Scholar Grant.

All authors are with the Division of Decision and Control Systems, EECS, KTH Royal Institute of Technology, Malvinas väg 10, 10044 Stockholm, Sweden {kfrag, frankji, kallej, jonas1}@kth.se. They are also affiliated with the Integrated Transport Research Lab and Digital Futures.

2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3315-0592-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ITSC58415.2024.10919551



(a) Pedestrian Trajectory Encoding (b) Clustering (c) Data-Driven Reachability Analysis

Fig. 2: **Framework Architecture** The figure displays the three main components of the proposed framework: **(a)** Trajectory Encoding Training Process, which involves the preparation and encoding of the input trajectories to extract informative features; **(b)** the Pedestrian Behavior Clustering for grouping the encoded embeddings based on their similarity; **(c)** Data-Driven Reachability Analysis of pedestrians using data of the same cluster

plex data relationships and enhancing modeling accuracy. While there is interesting work using transformers in time series prediction [11], [12] and improving the representation of time-series data [13], [14], to the extent of our knowledge, there is little previous work in the application of transformers for trajectory clustering. Thus, in this work, we explore this application and study the implications it has on data-driven reachability analysis for pedestrians. In Figure 1, we present an overview of our proposed approach that combines a transformer encoder with clustering and data-driven reachability analysis to predict the possible future states of pedestrians. The offline process involves training a transformer encoder to generate enriched trajectory representations, followed by clustering these encoded embeddings to create behavior clusters. Once deployed online, the system identifies the behavior cluster closest to the detected pedestrian and utilizes the data from this cluster to perform data-driven reachability analysis. By implementing this approach, we are able to utilize automatically identified behavior modes for data-driven reachability analysis.

A. *Contribution*

The main contribution of this paper is a pedestrian motion prediction framework that leverages transformers for clustering trajectories into behavior clusters, which are then used in data-driven reachability analysis to both reduce conservativeness and maintain safety. Explicitly, the contributions of the paper are three-fold:

1) we develop a framework that leverages transformers for clustering pedestrian trajectory data,
2) we present an integration of the resultant behavior clusters with data-driven reachability analysis,
3) we showcase the benefit of transformer-encoded trajectory clusters on data-driven reachability analysis using real-pedestrian trajectories.

Additionally, the code for the evaluations performed in this work is publicly available.[1]

The remainder of the paper is organized as follows. In Section II, we review the related work in trajectory clustering and transformer representation learning. In Section III, we describe our methodology for generating enhanced trajectory representations using transformers, clustering pedestrian behaviors, and implementing data-driven reachability analysis. In Section IV, we demonstrate our approach using real trajectory data. In Section V, we conclude the paper with a discussion and future directions.

II. Related Work

Due to the importance of pedestrian safety, there is an extensive body of literature exploring the use of behavior or intentions to improve the precision of pedestrian trajectory motion forecasting. For example, there are several recent proposals that incorporate pedestrian intentions to perform future trajectory predictions [7], [8], [15]. To support this work, several have developed a pre-defined set of labels and annotated datasets around pedestrian intentions and behaviors [16]–[20]. With a similar motivation, authors in [3] incorporate behavior modes into data-driven reachability analysis to further enhance the precision of the predicted future set of positions the pedestrian may occupy, assuming some level of decision uncertainty. The mentioned studies indicate the benefits of using behaviorally informed approaches for predicting the future motion of pedestrians. However, a remaining challenge with these approaches is the dependency on labels that are manually crafted based on human observations, which may not capture the full or accurate range of pedestrian behaviors and can introduce human bias.

[1] https://github.com/kfragkedaki/Pedestrian_Project

To alleviate this challenge, we explore the use of clustering for behavior labeling. Using clustering, we are able to automatically group similar pedestrian trajectories and, potentially, reveal behavioral patterns in historical data in an unsupervised manner. When treating historical pedestrian data as time series data, there are a variety of modifications for improving the clustering quality for trajectories [9], [10], [21]–[24]. Notably, authors in both [9], [10] adopt auto-encoder-based machine learning approaches for automatically generating new representations of trajectory data to improve the quality of the final clusters. By incorporating computations that convert historical trajectory data into higher-dimensional feature spaces, these approaches significantly improve the quality of the clustering output in a way that avoids the need for a human designer to manually perform labeling. Inspired by these works, we also seek to explore the use of machine learning for automatically converting historical pedestrian datasets into a feature space that yields better clusters.

In particular, we explore the use of transformer models, as an alternative to the auto-encoder-based models used in [9], [10], for enhancing trajectory clustering. Transformers, initially proposed for natural language processing [25], have been shown to capture long-range relationships and complex patterns through their attention mechanism, effectively process sequential data, and enable parallel computation. We are specifically motivated by the recent success in utilizing transformer models for improving regression and classification of time series data [13], capturing travel behavior and spatio-temporal correlations [14], and effectively encoding transportation-related road and route representations [26]. Although these applications of transformer models have shown great potential for encoding time-series and trajectory data for various tasks, the use of those models for enhancing trajectory clustering remains unexplored.

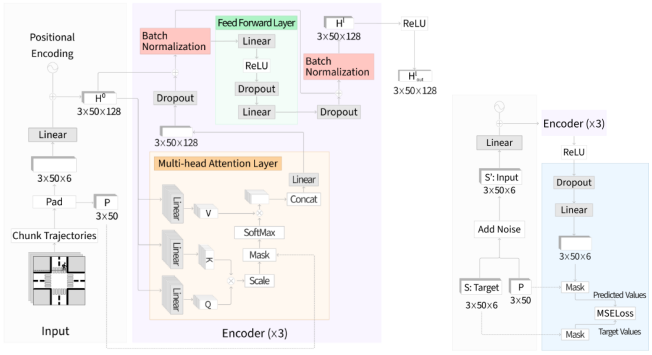
In this work, we explore an approach that leverages a transformer model to generate trajectory embeddings which can then be clustered and automatically yield behavior modes for improving pedestrian motion prediction. Specifically, we develop a framework that uses a transformer architecture similar to the one used in [13] to effectively cluster historical pedestrian trajectories. We consider each cluster as a "behavior mode" and, in the next sections, use the clusters to perform data-driven reachability analysis on pedestrians.

III. Methodology

The architecture of our approach is illustrated in Figure 2, and consists of the pedestrian trajectory encoding process, pedestrian behavior clustering, and data-driven reachability analysis based on the identified behavior clusters. Throughout the rest of this section, we will explain each of these parts in detail.

A. Pedestrian Trajectory Encoding

The first part of the system is the pedestrian trajectory encoding which refers to the unsupervised training of the trajectory encoder. In Figure 3, the transformer encoder and its training process are shown for three trajectories, and the details of the trajectory input preparation, the encoder model, and the training process are further explained below.



(a) Transformer Encoder (b) Training Process

Fig. 3: **Trajectory Encoding Model**: The figure illustrates the encoding process of three pedestrian trajectory instances segmented to 50 time points, with six features included at each time point: **(a)** Transformer Encoder of trajectory data; **(b)** the Autoregression Training Task for unsupervised training of the transformer encoder.

1) Trajectory Input Preparation: For this work, we utilize trajectory data consisting of the pedestrians' positions, velocities, and accelerations in both spatial dimensions, forming a 6-dimensional feature vector at each time point in the sequence. We develop our approach around these data points since this is the most commonly detected motion data by pedestrian detection systems. To address the variability in behaviors observed over longer trajectories, the data is segmented into smaller chunks of specified length d_c. We standardize the length of all input sequences by padding shorter trajectories with zeros, creating a binary matrix P to distinguish actual data from padded values in the model. Finally, each feature vector is projected from a 6-dimensional to a 128-dimensional space with positional encodings added to retain the sequential nature of the data, addressing the transformer architecture's inherent lack of input order consideration. The input to the transformer model is then the high-dimensional sequence matrix $H^0 \in \mathbb{R}^{d_b \times d_c \times 128}$, where d_b refers to the batch number of trajectory instances being processed and d_c to the chunk size of the trajectories.

2) Transformer Encoder: A transformer is a type of deep neural network that uses attention mechanisms to capture dependencies between elements in a sequence. In natural language processing (NLP), transformers are used to understand the context of a sentence, where the meaning of a word depends on its relationship with other words. For instance, the sentences "Where is the bus stop?" and "How does the bus stop?", despite containing similar words, have different meanings due to the order and relationships of the words. In our context, we implement the transformer architecture as described by [13], [25] to process pedestrian trajectory data. Similar to how words in a sentence have contextual relationships, each time point in a pedestrian's

trajectory is linked to preceding and succeeding movements, and by comparing them, features like the distance covered, the speed changes, or the walking style of the pedestrian can be inferred.

To perform this inference for pedestrian trajectories, we use the transformer encoder in Figure 3(a). As displayed in the figure, the higher dimensional trajectory embeddings H^0 are fed into a stack of three transformer encoder layers. Each layer comprises a pair of sublayers: a multi-head attention (MHA) and a fully connected feed-forward (FF) sublayer. We apply dropout [27] to both sublayers and then incorporate a skip connection [28] by adding the input of the sublayer to its output, followed by batch normalization (BN) [29]. The MHA sublayer utilizes a self-attention network with eight heads, enabling the model to capture different trajectory-related information simultaneously. On the other hand, the FF sublayer processes each position in the trajectory independently, applying the same weights across all positions to learn position-related information.

By using a stack of transformer encoder layers, the output embeddings from one layer serve as the input to the next layer, allowing the model to build upon the information learned in the previous layers. For example, if the model initially learns to identify basic characteristics such as distance covered and speed changes in early layers, then in the subsequent layers, it could use this foundational knowledge to infer more complex behaviors, such as determining whether or not a pedestrian is in a rush. This progressive learning and integration of information across layers enable a nuanced understanding of pedestrian dynamics that can be difficult for single-layer models to achieve. The high-dimensional embeddings extracted from the last layer of the encoder with ReLU activation applied, H^l_{out}, are the encoded trajectory embeddings used to identify behavior clusters in the next phases.

3) Autoregression Training Task: For training the encoder, we utilize an unsupervised learning approach based on the autoregression denoising task proposed by [13]. As shown in Figure 3(b), we add noise to the input data, by setting portions of it to zero, and task the model with predicting these masked values. The noised input is fed into the encoder, and the output embeddings of the trajectory encoder with dropout applied, H^l_{out}, are then mapped back to the 6-dimensional feature space. These outputs are compared with the original non-noised values to evaluate the model's predictions. We compute the Mean Squared Error (MSE) Loss, which measures the accuracy of predictions across all non-padded values, in contrast to [13] who focus solely on the masked data. This task aims for the encoder to learn from unlabeled data to generate informative trajectory embeddings.

B. Pedestrian Behavior Clustering

For clustering the encoded trajectory embeddings, we have chosen Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [30] due to its robustness in identifying clusters of arbitrary shapes and sizes. HDBSCAN does not require predefining the number of clusters, unlike other clustering algorithms such as K-means. Instead, it works by assessing the density of data points within a space and dynamically adjusting clusters based on a density threshold min_sample.

In the clustering process, the high-dimensional embeddings H^l_{out} extracted from the trajectory encoding phase are used. We apply mean pooling across the time dimension of these embeddings to aggregate the temporal information into a single representative vector for each trajectory. These pooled embeddings are then fed into the HDBSCAN algorithm to identify and group similar patterns of pedestrian behavior movements together.

C. Data-Driven Reachability Analysis

We perform data-driven reachability analysis of pedestrians using the clusters identified from the encoded trajectory embeddings. Our approach builds upon the work of [3], who used historically similar trajectories to predict future states. However, we enhance their framework by using automatically-identified behavior clusters instead of manually crafted labels. To predict the reachable set of a pedestrian, we first encode its trajectory using our trained transformer encoder. Then, we employ an Approximate Nearest Neighbor (ANN) algorithm to identify to which cluster the unseen trajectory most closely belongs. For brevity, we do not include the full formulation of the data-driven reachability analysis. For readers who would like to see the details of the reachability analysis, we direct them to [3, Algorithm 1], where the input $\mathcal{C}$ is the cluster of trajectories identified by ANN.

In this paper, we present the evaluation of only inputting the nearest cluster identified by ANN into the data-driven reachability analysis. Importantly, we note that the data-driven reachability analysis also supports the input of multiple clusters appended together. By increasing the number of clusters given to the data-driven reachability analysis, we increase both the conservativeness and safety of the predicted reachable sets. At the extreme, the inclusion of all historical data yields results similar to [5], which computes a reachable set that accounts for all possible linear models that are consistent with the historical data. Since the design of how many of the behavior clusters are selected is dependent on the particular application, we only evaluate the results of picking the closest one. An important future direction will be to develop schemes for adapting the number of chosen clusters based on the scenario.

IV. Evaluation

In this section, we present the trajectory encoding training and clustering results, and perform data-driven reachability analysis using the identified clusters. We also describe the dataset and experimental setup that were employed to conduct the evaluations presented.

A. Pedestrian Trajectory Dataset

We use the SIND dataset [31] to train and evaluate the transformer encoder and clustering on the work of [3]. The

SIND dataset is an open-source signalized dataset of real road users' trajectories from a large intersection in Tianjin, China, captured by a drone. We specifically focus on the pedestrian trajectories, which provide information about the spatial location, the velocity, and the acceleration over each time point in the trajectory. The location (x, y) is bounded by the spatial map limit of the dataset, and the velocity (v_x, v_y) and the acceleration (a_x, a_y) in both spatial dimensions are included. The trajectory data are filtered for falsely recognized pedestrians by completely removing stationary detections where all values of v_x and v_y in the trajectory equal zero.

As discussed in Section III, the data is segmented into chunks with a trajectory length of $d_c = 50$, and padded to standardize shorter trajectories. This differs from the work of [3] who use a chunk size of 90 and split the trajectories utilizing a sliding window technique of size one. By shortening the length of chunks and avoiding the sliding window approach, we aim to capture distinct pedestrian behaviors and reduce biases in the data. For training, we introduce noise to the input data following the mechanism introduced by [13] that randomly masks a proportion r of trajectory data based on a Bernoulli distribution with mean length l_m. To increase the complexity of the prediction task, both the mean length l_m of sequential input data masked and the proportion r of the data masked are incrementally increased during the training process.

The dataset is split into training, validation, and testing subsets, comprising 70%, 20% and 10% of the chunked data, respectively. The training and validation subsets are randomly split and used for training the transformer encoder. In the clustering phase, both the training and validation subsets are considered historical data, and are used to identify patterns and create the behavior clusters. The testing subset, which is fixed, is used exclusively for evaluating the effectiveness of the behavior clusters in the data-driven reachability analysis.

B. Experimental Setup

To evaluate the effectiveness of the proposed framework, we compare the following four approaches for selecting historical data to perform data-driven reachability analysis for a detected pedestrian.

1) **Baseline Method [3]**: We utilize all historical data that intersect with the detected pedestrian's position.
2) **Labeling Method [3]**: We split historical data into six predefined behavior modes, and the data of the mode of the detected pedestrian is utilized. We also apply a location and heading filter to the selected data.
3) **Non-encoded Trajectory Clustering**: We cluster historical data using the original 6-dimensional features and utilize all data in the detected pedestrian's cluster.
4) **Transformer-Encoded Trajectory Clustering**: We cluster historical data using encoded trajectory embeddings generated by the trained encoder of Figure 3(a). We utilize all data in the detected pedestrian's cluster.

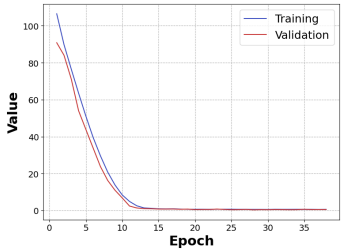


Fig. 4: Transformer encoder training and validation loss.

To ensure a fair comparison between these fundamentally different methods, we apply a cluster distance filter in both clustering techniques. Moreover, if the data provided for performing reachability analysis in any of the four methods exceeds the memory limitations of the machine running the analysis or if there is no data satisfying the above criteria, that trial is completely excluded. This is practically motivated, since in cases where memory is exceeded or no data is available, a fallback reachability analysis would be performed instead.

The implementation of our code is publicly available[2]. For the transformer-encoded trajectory clustering approach, we implemented our transformer encoder using PyTorch, and performed hyperparameter tuning with Ray, selecting parameters based on the MSELoss on the validation subset. Hyperparameter tuning was conducted on an Ubuntu system equipped with an Intel Core i7-8850H CPU at 2.60GHz, 12 GB of memory, and an NVIDIA GeForce RTX 2080 Mobile GPU. The chosen model was trained for 37 epochs using the Adam optimizer with a batch size of 256, a learning rate of 5.011×10^{-4}, L2 regularization of 0.05, a dropout rate of 0.1, and a ReLU activation function. To evaluate the learning curve of our model, we monitored MSE Loss on both the training and validation subsets. As shown in Figure 4, the MSE Loss decreases over time, indicating that the model is effectively learning to generalize to unseen data. For both clustering methods, we utilized the HDBSCAN algorithm from scikit-learn, while for the nearest neighbor cluster searches the Annoy[3] library.

C. Results

In this subsection, we present the results of our approach, showing the clusters created by both clustering methods and applying data-driven reachability analysis using all four aforementioned approaches.

1) Trajectory Clusters: After clustering both the initial trajectory data and the encoded trajectory embeddings, we applied Principal Component Analysis (PCA) to reduce the dimensionality for visualization and analysis. The initial trajectory data, consisting of position, velocity, and acceleration, was reduced from 6 dimensions to 3 dimensions, while the encoded trajectory embeddings were reduced from 128 dimensions to 3 dimensions. Figure 6 displays both

[2] https://github.com/kfragkedaki/Pedestrian_Project
[3] https://github.com/spotify/annoy

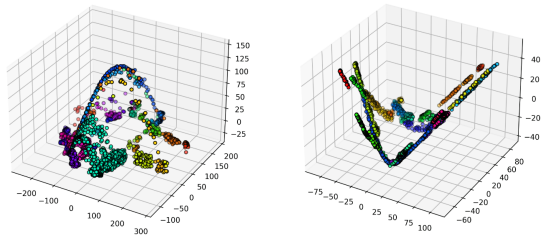
(a) Cross Illegal (b) Cross Now (c) Not Cross

Fig. 5: Reachable sets for three different scenarios using historical trajectories based on: the baseline and labeling oracle defined by [3], a non-encoded trajectory clustering, and a transformer-encoded trajectory clustering approach.

PCA-transformed data points, color-coded according to their respective clusters. We note that the PCA of the non-encoded initial data corresponds fairly closely to the geometry of the intersection, indicating a strong reliance on the position of the pedestrian for predicting its behavior. In contrast, the PCA of the encoded data shows a different structure, mapped to a narrower range of values. These visualizations emphasize that the encoded data differs from the initial trajectory data in structure and is picking up patterns in the data that are possibly more rich than just the position of the pedestrian.

D. Data-Driven Reachability Analysis Evaluation

We examine the reachability of pedestrian states using the identified transformer-encoded trajectory clusters and compare it with the other three methods. Figure 5 visualizes the reachable sets for three scenarios: (a) the pedestrian is crossing illegally (labeled "cross illegal"), (b) the pedestrian is crossing legally (labeled "cross now"), and (c) the pedestrian is not currently crossing (labeled "not cross"). In Figure 5(a), we see that the reachable set computed from the transformer-encoded trajectory cluster is clearly the most accurate and least conservative. In Figure 5(b), we see that the reachable set computed from the manually labeled data is the most accurate and least conservative, while the reachable set computed from the transformer-encoded trajectory cluster is comparable. This particular scenario provides a preliminary indication that, although a human can spend time manually creating clusters that may perform better, similar performance can be achieved in a fully automated fashion. In Figure 5(c) in which the pedestrian does not cross, we see that the transformer-encoded trajectory clustering approach is able to predict this behavior and generate an intuitive reachable set. Qualitatively, we can see that in these scenarios, the reachable sets generated from the transformer-encoded trajectory clusters could result in more efficient traffic flow, since vehicles can more precisely plan around the future motion of pedestrians. Quantitatively, based on the volume of each method in the scenarios and the average volumes on the testing data presented in Table I, we can conclude that the baseline method is clearly overly conservative, while the transformer-encoded trajectory clustering method is competitive with the labeling approach, while outperforming the non-encoded trajectory clustering.



(a) Initial Data (b) Encoded Data

Fig. 6: PCA on the initial data and the encoded trajectory embeddings, color-coded by cluster.

TABLE I: Comparison of zonotope area volumes in the scenarios, and the average zonotope volumes on the testing dataset.

	Baseline (m^2)	Labeling (m^2)	Non-encoded (m^2)	Transformer-encoded (m^2)
Cross Illegal	657.0511	122.0815	204.1029	**73.8813**
Cross Now	454.6648	**98.4315**	-	172.4226
Not Cross	396.1144	141.6194	169.2804	**52.6826**
Average Volume	309.2136	175.6101	320.1112	259.526

Additionally, to assess the overall safety of each method, we evaluate each method's state inclusion accuracy, shown in Figure 7. This metric represents how frequently the actual future state of a pedestrian falls within the predicted reachable set. The transformer-encoded trajectory clustering approach achieves the best performance after the overly conservative baseline, while the non-encoded trajectory clustering performs the worst out of all four methods. For evaluating these results, impractical outlier cases needed to be fully removed from the experiments. For brevity, we do not report the statistics of these outlier cases and direct interested readers to the open-source implementation for more details.

V. Discussion & Future Work

In this work, we propose a transformer-encoded trajectory clustering approach that automatically selects historical trajectories for data-driven reachability analysis of pedestrians. This approach is shown to be effective in maintaining safety while enhancing the precision of pedestrian motion predictions. The transformer-based method offers a clear ad-

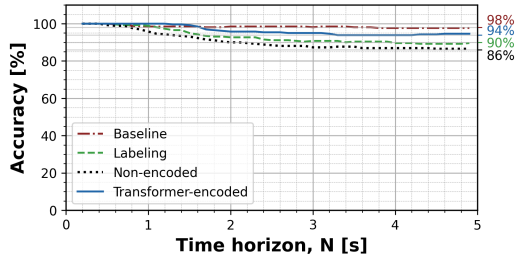


Fig. 7: State inclusion accuracy on the testing dataset.

vantage over classical clustering by capturing complex data relationships and improving modeling accuracy. From our findings, we observe that the reachability analysis can face challenges due to either insufficient or excessive historical data, and there is more work to be done on how to safely handle these cases. Finally, since the SIND dataset captures pedestrian motion from the perspective of a drone, evaluating our approach on data captured from a vehicle's perspective is important to understand its applicability in automated vehicles. This evaluation will help determine the feasibility and effectiveness of our transformer-encoded trajectory clustering in real-world scenarios involving autonomous driving.

REFERENCES

[1] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[2] C. Pek, S. Manzinger, M. Koschi, and M. Althoff, "Using online verification to prevent autonomous vehicles from causing accidents," *Nature Machine Intelligence*, vol. 2, no. 9, pp. 518–528, 2020.

[3] A. Söderlund, F. J. Jiang, V. Narri, A. Alanwar, and K. H. Johansson, "Data-driven reachability analysis of pedestrians using behavior modes," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023.

[4] A. Devonport, F. Yang, L. El Ghaoui, and M. Arcak, "Data-driven reachability analysis with christoffel functions," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 5067–5072.

[5] A. Alanwar, A. Koch, F. Allgöwer, and K. H. Johansson, "Data-driven reachability analysis from noisy data," *IEEE Transactions on Automatic Control*, 2023.

[6] A. Alanwar, F. J. Jiang, M. Sharifi, D. V. Dimarogonas, and K. H. Johansson, "Enhancing data-driven reachability analysis using temporal logic side information," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6793–6799.

[7] S. Malla, B. Dariush, and C. Choi, "Titan: Future forecast using action priors," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 183–11 193.

[8] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, in *Peeking into the future: Predicting future person activities and locations in videos*. IEEE Computer Society, Jun 2019, pp. 5718–5727.

[9] D. Yao, C. Zhang, Z. Zhu, J. Huang, and J. Bi, "Trajectory clustering via deep representation learning," vol. 2017-May. Institute of Electrical and Electronics Engineers Inc., Jun 2017, pp. 3880–3887.

[10] X. Olive, L. Basora, B. Viry, R. Alligier, and B. Viry, "Deep trajectory clustering with autoencoders," Sept 2020.

[11] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 10 335–10 342.

[12] L. Franco, L. Placidi, F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Under the hood of transformer networks for trajectory forecasting," *Pattern Recogn.*, vol. 138, no. C, jun 2023.

[13] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, ser. KDD '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 2114–2124.

[14] Y. Lin, H. Wan, S. Guo, J. Hu, C. S. Jensen, and Y. Lin, "Pre-training general trajectory embeddings with maximum multi-view entropy coding," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–15, 12 2023.

[15] A. Rasouli, I. Kotseruba, T. Kunic, and J. Tsotsos, "Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6261–6270, Oct 2019.

[16] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 206–213.

[17] G. Awad, A. A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, D. Joy, A. Delgado, A. F. Smeaton, Y. Graham, W. Kraaij, G. Quénot, J. Magalhaes, D. Semedo, and S. Blasi, "Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search," in *Proceedings of TRECVID 2018*, 2018.

[18] M. S. Aliakbarian, F. S. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson, "Viena^2 : A driving anticipation dataset," in *Computer Vision – ACCV 2018*. Springer International Publishing, 2019, pp. 449–466.

[19] B. Liu, E. Adeli, Z. Cao, K.-H. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles, "Spatiotemporal relationship reasoning for pedestrian intent prediction," in *IEEE Robotics and Automation Letters (IEEE RA-L) and International Conference on Robotics and Automation (ICRA)*, 2020.

[20] T. Chen, T. Jing, R. Tian, Y. Chen, J. Domeyer, H. Toyoda, R. Sherony, and Z. Ding, "Psi: A pedestrian behavior dataset for socially intelligent autonomous car," Dec 2021.

[21] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," *Proceedings - International Conference on Data Engineering*, pp. 673–684, 2002.

[22] J. G. Lee, J. Han, and K. Y. Whang, "Trajectory clustering: A partition-and-group framework," 2007, pp. 593–604.

[23] C. Sung, D. Feldman, and D. Rus, "Trajectory clustering for motion prediction," 2012, pp. 1547–1552.

[24] S. Wang, Z. Baoy, J. S. Culpeppery, T. Sellisz, and X. Qinx, "Fast large-scale trajectory clustering," vol. 13. VLDB Endowment-PUB4722, Sept 2019, pp. 29–42.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[26] Y. Chen, X. Li, G. Cong, Z. Bao, C. Long, Y. Liu, A. K. Chandran, and R. Ellison, "Robust road network representation learning: When traffic patterns meet traveling semantics," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ser. CIKM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 211–220.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, p. 448–456.

[30] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172.

[31] Y. Xu, W. Wang, H. Huang, H. Wang, W. Shao, J. Li, K. Yang, and C. Lv, "Sind: A drone dataset at signalized intersection in china (a) (b)."