Pedestrian motion prediction evaluation for urban autonomous driving

Dmytro Zabolotnii[1], Yar Muhammad[2], and Naveed Muhammad[1]

***Abstract*—Pedestrian motion prediction is a key aspect in any autonomous-driving pipeline, and is required for ensuring safe, accurate, and timely awareness of human agents' possible future trajectories. Autonomous vehicles need to use agent motion-prediction information to prevent any possible accidents, and for creating a comfortable and pleasant driving experience for vehicles' passengers as well for pedestrians in vehicles' vicinity. A significant amount of research has been conducted on the topic of agent motion prediction in the fields of robotics, computer vision, and intelligent transportation systems etc. However, a relatively unexplored aspect in the existing literature is the integration of state-of-the-art motion-prediction solutions into existing autonomous driving stacks and evaluating them in real-life conditions rather than on sanitized datasets. In this paper, we analyze a set of selected methods from the literature, and present the perspective obtained by integrating them into an existing autonomous-driving software stack – Autoware Mini – and performing experiments in natural urban conditions in Tartu, Estonia to determine the suitability of conventional motion prediction metrics. Our study should be of value to researchers in autonomous driving or robotics interested in understanding real-world performance of existing state-of-the-art pedestrian motion prediction methods. The code, along with instructions on accessing the dataset that we employ, is available at https://github.com/dmytrozabolotnii/autoware_mini**

***Index Terms*—Autonomous Vehicle Navigation, Computer Vision for Transportation, Datasets for Human Motion, Human Detection and Tracking, Intention Recognition**

I. Introduction

Autonomous driving research is an exciting topic nowadays, with an aggrandizing promise to improve the driving process for everyone and eventually replace human drivers, starting from future long-distance truck operators [1] to already running autonomous taxi services [2]. However, even if human drivers can be replaced, the human factor will never disappear from autonomous driving. One of the most significant human factors is not even connected to the vehicle itself - but rather to the agency of other human actors on the road - other drivers in non-self-driving automobiles and pedestrians [3]. A self-driving vehicle should be aware of their possible future actions and apply necessary corrections to its course of action to avoid collisions, follow legal directives, and ensure a safe and comfortable environment for all actors. This function is essential for ensuring pedestrian safety. Poor interactions between vehicles and pedestrians lead to many traffic accidents, with over 80% possibility of the fatal outcome when the vehicle moves over 60 km/h [4]. This issue is even more exorbitant among youth and elderly groups, especially in developing countries, with road-caused accidents being the lead cause of youth disabilities globally [5].

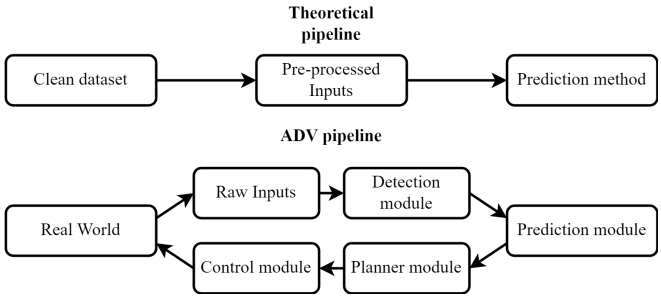

Fig. 1. Architectural difference between implementation of state-of-the-art motion prediction methods and their potential implementation in a modular Autonomous Driving framework.

Accordingly, to solve these problems in future autonomous driving applications, pedestrian motion prediction remains an active research topic, with numerous new solutions published every year. Starting from the classic physics-based models [6], the trends shifted to Machine-Learning-based solutions over the last decade [3]. In the last few years of the published articles, there is a wide representation of underlying architectures used for prediction [7]: Convolutional Variational Autoencoder (CVAE) methods [8]–[11], Generative Adversarial Network (GAN) methods [12]–[14], Transformer methods [15]–[17], Diffusion methods [18], [19], and even as novel application of Large Language Model-based methods [20], [21]. However, while the variety of underlying methods is excellent, the evaluation of these methods is not as outstanding. The majority of newly published (that often claim to be state-of-the-art) use Average Displacement Error/Final Displacement Error (ADE/FDE) metrics for validation (or their minADE/minFDE variants), despite emerging research about deficiencies of these metrics [22]–[24] and the existence of appropriate alternative metrics that have been proposed in the literature, such as negative log-likelihood (NLL) or Average Mahalanobis Distance (AMD) [25]. Additionally, model training and experimental evaluation are often done

[1] D. Zabolotnii and N. Muhammad are with the Institute of Computer Science at the University of Tartu, Narva mnt 18, Tartu 51009, Estonia. Email: dmytro.zabolotnii@ut.ee, naveed.muhammad@ut.ee

[2] Y. Muhammad is with the Department of Computer Science at the University of Hertfordshire, AL10 9AB Hatfield, U.K. Email: y.muhammad@herts.ac.uk

This work was supported in part by European Social Fund through the "ICT Programme" Measure and in part by Bolt Technologies through the Collaboration Project under Grant LLTAT21278.

only on small and somewhat outdated datasets such as ETH/UCY [26], [27] and SDD [28] instead of more extensive and much more diverse datasets such as Argoverse [29] or Waymo Open Motion [30].

Even more fundamentally, there remains an inconsistency between evaluating any prediction method on the offline dataset and evaluating the prediction method as the module in the modular Autonomous Driving framework, as illustrated by Fig. 1. During the application of the prediction algorithm as part of the modular framework, not only is it affected by the potential errors in the upstream detection and sensor modules, but it also influences the planning and control modules that lead to the movement of the ego-vehicle, and correspondingly all the exo-agents whose motion is predicted can change their future actions because of our movements. This compounding error effect leads to what is called *dynamics gap* in [23]. Correspondingly, the dynamics gap is absent from the traditional motion prediction datasets, as both the input and ground truth data undergo refinement by human labelers and other pre-processing techniques, considerably reducing measurement errors. These techniques are often not applicable in real-time experimental environments due to execution time constraints or the requirement to have data from future points in time. The discrepancy in evaluation, combined with an outdated approach to metrics and existing datasets, creates the primary motivation for this study.

In this paper, we aim to expand upon the evaluation of existing state-of-the-art methods using a more realistic experimental setup - an open-source autonomous driving framework with real-life sensor data. The necessity of processing raw sensor data and running all the other parts of the autonomous driving framework simultaneously as the prediction algorithms on the limited consumer-grade hardware that is expected to be in the autonomous vehicle requires stringent evaluation of the computational efficiency. At the same time, we want to challenge the standard Best-of-*N* approach (where the prediction algorithms are allowed to output *N* predictions, but only the best one is assessed) that uses ADE/FDE metrics for evaluating the prediction algorithm in the existing pedestrian motion prediction datasets benchmarks. As such, our contributions are three-fold:

1) Evaluation of the pre-selected pedestrian motion methods and a baseline method inside the Autoware Mini framework under different output modalities, creating the unique perspective of state-of-the-art prediction methods performance.
2) Engineering adaptation of chosen prediction methods to enable their online performance inside an autonomous driving framework.
3) Creation of an experimental dataset from raw data recorded during past autonomous vehicle trips in Tartu, Estonia.

II. Research Methods

A. Problem Formulation

We approach the pedestrian motion prediction problem primarily as the problem of future trajectory prediction based on historical observations, or, as they are sometimes referred, stimuli [3]. There are several commonly used stimuli however the most prominent is the previous locations of the pedestrian agents represented in 2D from a Birds-Eye-View (BEV) perspective [7]. As such, we can formalize the problem of prediction in the following way: given the historical trajectories of N pedestrians over H previous states $X_i = x_i^1, x_i^2, ..., x_i^H$, find the future trajectories of the pedestrians over F future states $Y_i = y_i^{H+1}, y_i^{H+2}, y_i^{H+F}$, or in another form, find function f that satisfies $Y_i = f(X_i)$ [31].

This simple problem formulation is used when introducing physics-based solutions, such as the Constant Velocity Model (CVM) [6]. However, it is insufficient for the more recent data-driven approaches. To extend it, we can represent x_i^j as not only the physical location of the pedestrian p_i^j, but as a collection of location and other auxiliary inputs $x_i^j = (p_i^j, a_i^j, b_i^j, ...)$ such as environment representation (HD-map), inner characteristics of the pedestrians (gestures, emotions) or raw sensor data. Then, data-driven approaches need to estimate model M that represents output trajectory Y_i given input features of all N pedestrians to represent cross-actor interactions: $Y_i = M(X_i, \{X_j\}_1^N)$. Often, instead of direct deterministic output, the model represents probability distribution $p(Y_i|X_i, \{X_j\}_1^N)$ from which multiple candidate trajectories can be sampled, with the intent that at least one of them will be close to future ground-truth.

B. Motion prediction Algorithms

The main step in selecting state-of-the-art pedestrian motion prediction methods is finding them. The original search was performed in three distinct steps: first, the relevant keywords search was performed on academic journals to find corresponding survey articles [3] [31] [7] covering latest advancements in the field; second - filtering out articles of interest from all reviewed by reading the original publication and finding if they fulfill necessary criteria; third - analyzing the open source solution of algorithms and cross-referencing the published results with benchmarks of the used datasets. The criteria for filtering out the articles are as follows:

1) The model is based on a supervised learning algorithm
2) Pre-trained model allows real-time inference on the limited hardware constraints.
3) The model's main input should be the historical trajectories of the pedestrians. While many models rely on other input data, such as HD-map, in practice, it is hard to transfer learned features from the HD-map

TABLE I
Overview of chosen methods

Model	Year	Input	Cross-agent interaction consideration	Output Modality
PECNet	2020	Trajectory	Yes	Stochastic
SGNet	2022	Trajectory	No	Stochastic
GATraj	2023	Trajectory	Capable	Probabilistic
MUSE	2022	Trajectory+Map	No	Probabilistic
CVM	-	Final Velocity	No	Deterministic

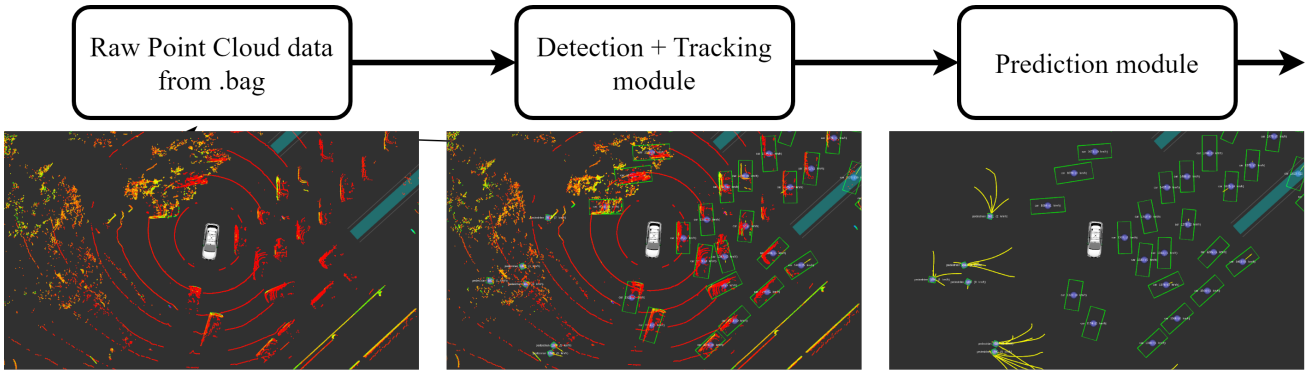

Fig. 2. Data flow inside Autoware Mini framework. Detection module extracts the shapes of the objects from raw point cloud, and classifies them to pedestrian/vehicle/other objects. Afterwards, selected prediction model outputs candidate trajectories, represented here as yellow curves

used for the training to the HD-map used in different autonomous stacks.

4) Open source implementation of the model's architecture, preferably with access to pre-trained weights that reproduce results published in the article. While open-source implementation distributions are common, they are often incomplete, with details missing both from the training code and the description of the training process in the original publication.

Following these criteria, four methods from the last four years were chosen: a) PECNet [32] b) SGNet [10] c) GATraj [17] d) MUSEVae [8]. Furthermore, we establish CVM as a useful baseline that is proven to be comparable and even outperforms data-driven approaches under certain conditions [33]. CVM is a simple physics-based model that assumes that objects will continue moving along their estimated velocity vector, disregarding acceleration and other factors. The overall comparison of methods according to our problem formulation is presented in Table I. While we specified that reliance on HD-map is a negative criteria, MUSE-VAE method uses the semantic map as an auxiliary input with very few encoding categories that allow reliable adaptation of our existing mapping data. GATraj can integrate interactions between separate pedestrians, but the ablated version of the model (dropping the GCN module) without this feature was used due to hardware constraints. Finally, stochastic output modality denotes that corresponding models output multiple predicted trajectories at once, however, models do not assign a numerical probability to output trajectories, while probabilistic models both output multiple trajectories and assign the probability, making it possible to choose the most likely trajectory(ies) for the evaluation.

C. Experimental Setup

1) Data collection: Our goal is the evaluation of the existing prediction models in the environment as close as possible to a real autonomous vehicle. As such, we base our setup on Autoware Mini framework [34], a lightweight fork written in pure Python of the original Autoware.AI [35], one of the first modular autonomous driving frameworks, which itself is based on Robot Operating System (ROS) [36]. This framework exists both as a scientific and a pedagogical tool, but it has also been tested on Lexus RX450h vehicle on the streets of Tartu, Estonia. The vehicle is equipped with Ouster OS1-128 and Velodyne VLP-32C lidars as the primary sensors for object detection. All data these and other sensors recorded during regular autonomous and manual trips for the last three years are stored inside .bag file format. This raw sensor data collection provides a primary dataset for our evaluation purposes. However, the entire data collection is 3400+ different .bag files, totaling over 43 TB, so some filtering was necessary to produce a compact and interesting dataset. The primary filters were the pedestrians' rich presence, long uninterrupted driving periods, and small-to-no weather interruptions to make sensor readings as clear as possible. The final result was a dataset with 18 .bag files, where every .bag file represents a continuous scene, with a total runtime of 4 hours and 15 minutes, which is comparable to other popular datasets that include raw sensor data, such as Nuscenes (6 hours) [37] and Open Waymo Perception (11 hours) [30]. All selected .bag files were recorded before implementation of the selected predictor models and cover a variety of scenarios: normal city traffic inside Tartu city, busy city central intersection with dense pedestrian traffic, and prolonged trips from city center to the city suburb.

2) Implementation Details: Using .bag files inside Autoware Mini framework allows to replay them, simulating the stream of data that the actual autonomous vehicle received during the trip and allowing to run an entire framework processing the data through autonomous driving architecture described in Figure 1. This data flow is represented in the Figure 2. Bag files output raw lidar point cloud data to the detector module. For detection, our framework uses an SFA3D detector based on [38] to process raw point cloud data and output labeled vector representation of the objects. Next, the simplistic exponential moving averages-based tracker [39] keeps the object's permanence that allows the construction of historic pedestrian trajectories in real time. Finally, the prediction module receives historical trajectories and other auxiliary data and outputs candidate trajectories that are forwarded to the rule-based planner module of Autoware Mini. While the planner module reacts to the predicted

trajectories and directs controls of the vehicle, this does not affect the behavior of the exo-agents as it is a replay of the recorded sensor data.

However, because of the errors introduced by these upstream algorithms, the raw dataset is unsuitable for chosen prediction models training, which necessitates the usage of the pre-trained models. For our evaluation, we use PECNet and MUSE-VAE pre-trained on SDD [28], while SGNet and GATraj are pre-trained on ETH/UCY [26], [27]. Both datasets follow the same input format of 8-point historical trajectories to predict a 12-point candidate trajectory, where points are 0.4 seconds apart. As a result, all models consider 3.2 seconds of past data to predict the next 4.8 seconds of motion. Consequently, all models were adapted to run with a constant stream of incoming data, where every 0.4 seconds, they receive updated trajectories of all visible pedestrians. As such, the key requirement for performance was that the inference with a variable batch size finishes in 0.4 seconds before the next set of trajectories is obtained. Despite this interval limiting the effective performance of the predictor module to 2.5 Hz, it does not slow down the performance of the framework that runs at 10 Hz at minimum to ensure real-time vehicle reactions.

To simulate the limited resources of an embedded system of the real vehicle, the complete evaluation framework was performed on a consumer-grade laptop equipped with one NVIDIA GeForce 3080 and AMD Ryzen 5900HX. Apollo [40] uses a similar hardware setup in already deployed autonomous vehicles. This hardware severely limits performance compared to how these prediction models are usually evaluated - using multiple consumer-grade GPUs or GPUs specifically created for deep learning tasks like NVIDIA V100/A100.

D. Evaluation Metrics

The most commonly used metrics for evaluating trajectory prediction are Average Displacement Error (ADE) and Final Displacement Error (FDE) [3]. ADE metric is commonly defined as the average L2 distance between all points of predicted trajectory and ground truth future trajectory:

$$ADE = \frac{1}{N}\frac{1}{F}\sum_{i=1}^{N}\sum_{j=1}^{F}||y_i^{H+j} - \hat{y}_i^{H+j}||_2$$

, where $\hat{Y}_i = \hat{y}_i^{H+1}, \hat{y}_i^{H+2}, ..., \hat{y}_i^{H+F}$ is the ground truth future trajectory. FDE is similar, but considers only L2 distance between final points of predicted and ground-truth trajectories.

$$FDE = \frac{1}{N}\sum_{i=1}^{N}||y_i^{H+F} - \hat{y}_i^{H+F}||_2$$

However, many data-driven approaches output multiple candidate trajectories. To solve this, the variations of ADE/FDE were introduced called minADE/minFDE, which calculates metrics for every candidate trajectory but considers only the best (lowest) metric score. This variant is commonly used for benchmarking on most pedestrian motion prediction datasets, such as SDD, ETH-UCY, Nuscenes, and Argoverse, with the only difference being that dataset evaluation restricts the amount of candidate trajectories methods that can be submitted. Early datasets such as SDD and ETH-UCY allow up to $k = 20$ candidate trajectories, while Nuscenes has two different benchmarks for $k = 5$ and $k = 10$ and Argoverse v1/2 settling down for $k = 6$.

The methodology of using only best score (Best-of-*N* approach) has been recently called into question [22], [23], [25], [41], [42]. The central point of the critique is that if the future ground truth trajectories are parametric distribution (Gaussian or mixed-Gaussian in the simplest case), and any model tries to estimate this distribution using minADE/minFDE, it instead estimates the square root of PDF [41], which means that it will always perform worse than Bayes-optimal predictor. The theoretical way to recover from this is to increase the amount of k trajectories we sample significantly. However, this approach is impractical in autonomous driving, where there are performance constraints to obtaining more sample trajectories if our inference time is long, and a large number of predicted trajectories can cause a problem for downstream planning and control modules (for example, a pedestrian is moving on the sidewalk, and low probability candidate trajectory falsely predicts they will cross a road, forcing the vehicle to react, when with lower amount of candidate trajectories this prediction would not exist). Another point of the criticism is that existing metrics poorly represent the dynamics gap [23] - the idea is that as much as our autonomous vehicle planner considers predicted trajectories and reacts to them, the agents whose motion we predict also consider and predict our ego-vehicle movements, creating dependency on each others' predictions. To resolve this, we follow [23], and implement Dynamic Average Displacement Error (DynADE) and Dynamic Final Displacement Error (DynFDE) metrics. The critical difference is the interactivity factor - in standard ADE/FDE calculation, we calculate metric once per given trajectory in the dataset, while in our setup, as we receive the updated trajectory of each agent in the online stream of data, we calculate the new metric value with every newly added trajectory point, averaging these L2 distance values for every agent, and then averaging over all agents at the end of each scene. Formally:

$$DynADE = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{L_i}\sum_{j=1}^{L_i}\frac{1}{F}\sum_{k=1}^{F}||y_i^{j+k} - \hat{y}_i^{j+k}||_2$$

$$DynFDE = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{L_i}\sum_{j=1}^{L_i}||y_i^{j+F} - \hat{y}_i^{j+F}||_2$$

,

where L_i is the total length of agent i trajectory in the observed scene. We average results from each .bag file scene for our dataset to obtain the final result. The new metric is similar to standard ADE/FDE but correlates more significantly with driving performance when integrated into the autonomous driving framework [23]. Correspondingly, for models that output multiple candidate trajectories, Best-of-*N*

TABLE II
QUANTITATIVE RESULTS WITH VARYING AMOUNT OF CANDIDATE TRAJECTORIES k

Model	CVM	PECNet			SGNet			GATraj			MUSE-VAE		
k	1	1	5	10	1	5	10	1	5	10	1	5	10
.bag dataset													
minDynADE (m)	1.605	1.749	1.637	1.564	1.845	1.390	1.297	1.871	1.133	**0.918**	2.587	1.909	-
minDynFDE (m)	2.970	3.276	3.021	2.854	3.476	2.649	2.458	3.589	2.012	**1.563**	5.212	3.739	-
Nuscenes													
minADE (m)	3.70	-	-	-	-	1.85	1.32	-	1.87	1.46	-	1.38	**1.09**
minFDE (m)	9.09	-	-	-	-	3.87	2.51	-	4.06	2.97	-	2.90	**2.10**
ETH/UCY (k=20)													
minADE (m)	0.52	0.29			0.18			**0.17**			-		
minFDE (m)	1.14	0.48			0.35			**0.29**			-		

TABLE III
ABLATION TESTING WITH VARYING AMOUNT OF CONSIDERED HISTORICAL STEPS H ON .BAG DATASET

Model	CVM	PECNet			SGNet			GATraj			MUSE-VAE		
H	0	2	4	8	2	4	8	2	4	8	2	4	8
minDynADE (m)	1.605	2.043	1.824	1.637	2.194	1.479	1.390	1.281	1.165	1.133	1.910	1.964	1.909
minDynFDE (m)	2.970	3.762	3.382	3.021	3.661	2.807	2.649	2.268	2.049	2.012	3.862	3.861	3.739

approach is still used by choosing the candidate trajectory with the lowest metric score to allow comparison between performance on our dataset and standard datasets such as Nuscenes, denoted as minDynADE/minDynFDE.

III. FINDINGS

A. Quantitative results

We first present a quantitative evaluation of the chosen algorithms and baseline over a created dataset in Table II. We also provide evaluation of the selected methods on the Nuscenes and ETH/UCY datasets for comparison, sourced from original papers [8], [10], [17], [32] (PECNet is not evaluated on the Nuscene dataset, and MUSE-VAE is not evaluated on ETH/UCY). To demonstrate the effect of the Best-of-*N* approach, we test selected methods under variable candidate trajectories amount $k = 1, 5, 10$. If the model provides more trajectories than k during the single inference run, then k trajectories were selected randomly unless the specified method also outputs the probability of every candidate trajectory, in which case k most likely trajectories were selected. As previously stated, all methods except baseline CVM rely on trajectories over the past 3.2 seconds while outputting candidate trajectories for 4.8 seconds, in line with pre-trained model original datasets ETH/UCY and SDD. Nuscenes results are instead obtained using past 4 seconds to predict next 6 seconds. Several insights can be derived from the results:

- Inline with theory, Best-of-*N* metric of every data-driven model increases with increase of k. However, increasing k beyond 10 is impractical in the framework due to performance limitations and potential issues in downstream modules such as planning.
- Given $k = 1$, none of the models outperform the Constant Velocity Model despite considering more than the final state of the pedestrian trajectory. Even with the increase of k, machine learning methods do not outperform the baseline to the degree expected from the previously presented results on the Nuscenes and ETH/UCY offline datasets. Considering that CVM is a far simpler and faster predictor, this shows that CVM remains a viable alternative approach in time-critical applications. These findings align with previous research [33] and demonstrate that pedestrian motion prediction remains an unsolved problem under certain conditions, opening avenues for future research that needs to consider other input modalities instead of relying on trajectory only.
- MUSE-VAE model, despite being the only model that relies on additional auxiliary map input data, performs the weakest in our framework. The result shows that transferring learned map features from one dataset to another is non-trivial, even for the simple semantic map representation. Additionally, MUSE-VAE implementation wasn't able to output $k = 10$ candidate trajectories under the required time constraint of 0.4 seconds, leading to no results available for Table II.
- Performance of the selected methods on the .bag dataset correlates with the performance of the methods on ETH/UCY dataset, which contains only pedestrians. However, on the Nuscene dataset, MUSE-VAE outperforms other methods, which is not replicated in our evaluation. This result confirms that while learning map representation on the known dataset (where locations of train/test split samples are likely to overlap) is highly beneficial, it does not transfer well to unknown map configurations with different geography.

B. Ablation testing

The critical issue for pedestrian motion prediction in the context of autonomous driving is the limited availability of historical information. During testing on our dataset, it is common that some pedestrians are only detected very close to the ego-vehicle, giving no time to construct an entire 3.2-second trajectory history before the critical prediction

needs to be made. As such, we perform ablation testing where we limit the amount of historical trajectory points models consider H to lower value, simulating the lower length of historical trajectory, and compare with the results on default behavior $H = 8$ in table III (for all methods except CVM $k = 5$). From this testing, it can be seen that while PECNet gradually improves its predictions while incorporating more historical trajectory steps, GATraj and SGNet plateau performance on $H = 4$, and Muse-VAE seems to rely almost entirely on auxiliary map information and last $H = 2$ trajectory points, as no meaningful improvements observed with increase in H. The testing shows that while models claim that they consider entire past trajectories, they do not always encode useful information from the oldest trajectory steps.

C. Limitations

Due to the internal workings of ROS adding delays during data transfer between different framework modules, it is impossible to guarantee deterministic metrics calculation on a specific .bag scene. This is caused not only by distinct predictions generated by prediction modules every replay of the .bag scene but also by potential deviation of the ground truth against which metrics are calculated due to errors introduced by upstream detection and tracking modules. As such, we execute one scene evaluation 20 times using every predictor (at $k = 5$ and $H = 8$) and calculate the standard deviation and mean of the obtained metric values to showcase possible deviation ranges from the published results in this section. The results are available in Table IV. The calculated margin of error from +-3% to +-6% for different methods is deemed acceptable for interpretations of the general evaluation results, but the approach used does not allow to distinguish and calculate specific errors introduced by the upstream modules from other factors.

TABLE IV
POSSIBLE VALUES OF STANDARD DEVIATION DURING REPRODUCTION OF THE RESULTS

Model	MinDynADE		MinDynFDE	
	Mean	Std	Mean	Std
PECNet	1.427	0.049	2.615	0.096
SGNet	1.081	0.061	2.056	0.118
GATraj	0.935	0.044	1.603	0.084
MUSE	1.484	0.074	2.808	0.136
CVM	1.451	0.051	2.724	0.090

The presented results make a strong case for the evaluability of CVM and insufficiencies of tested Machine-Learning models' however contextual information about .bag dataset cannot be generalized to other environments. The trips were recorded inside Tartu, a relatively small city where massive moving crowds of people are a rare occurrence and, as such, often contain completely static agents. In this scenario, Machine-Learning models perform sub-optimally in comparison to more dynamically and non-uniformly moving agents [33]. Additionally, the implemented prediction modules currently consider interactions between pedestrians only and not with other traffic agents, such as vehicles or important static objects like traffic lights or signs that govern pedestrian movement in complex urban scenarios. Accounting for these relationships while not treating pedestrians and vehicles as homogeneous agents is an important open avenue for research.

IV. CONCLUSION

In this work, we have performed a comprehensive evaluation of selected state-of-the-art pedestrian motion prediction methods on the dataset obtained from an actual autonomous vehicle trips in the software environment of the autonomous driving framework under similar hardware restraints. Our evaluation focused on testing limits of the Best-of-*N* approach of the standard ADE/FDE metrics using modified implementation, created to resolve issues arising from implementing standard solutions into the full modular autonomous driving stack. We found that none of the evaluated models performed better than the standard Constant Velocity Model when outputting only a single candidate trajectory. Otherwise, GATraj model performs best, both when increasing the number of candidate trajectories beyond one and decreasing the length of the historical trajectory input observation window.

The study confirms that pedestrian motion prediction remains an open problem under specific conditions and that researchers should more strictly evaluate the potential practical application of their research. Many better or more promising machine-learning models were not evaluated in this study due to hardware or implementation constraints. As such, future research should focus on more lightweight practical solutions to the problem while also developing methods of incorporating input data other than historical trajectories and accounting for complex interactions between pedestrians and other traffic agents. We hope our research is helpful to scientists working on new pedestrian motion prediction models and engineers working on autonomous driving vehicles.

REFERENCES

[1] A. Herrmann, W. Brenner, and R. Stadler, *Autonomous driving: how the driverless revolution will change the world*. Emerald Group Publishing, 2018.

[2] "Dmv approves cruise and waymo to use autonomous vehicles for commercial service in designated parts of bay area," Sep 2021. [Online]. Available: https://www.dmv.ca.gov/portal/news-and-media/117199-2/

[3] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[4] C. V. Zegeer and M. Bushell, "Pedestrian crash trends and potential countermeasures from around the world," *Accident Analysis & Prevention*, vol. 44, no. 1, pp. 3–11, 2012.

[5] C. Branche, J. Ozanne-Smith, K. Oyebite, and A. A. Hyder, "World report on child injury prevention," 2008.

[6] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020.

[7] E. Schuetz and F. B. Flohr, "A review of trajectory prediction methods for the vulnerable road user," *Robotics*, vol. 13, no. 1, p. 1, 2023.

[8] M. Lee, S. S. Sohn, S. Moon, S. Yoon, M. Kapadia, and V. Pavlovic, "Muse-vae: Multi-scale vae for environment-aware long term trajectory prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 2221–2230.

[9] H. Zhou, D. Ren, X. Yang, M. Fan, and H. Huang, "Csr: cascade conditional variational auto encoder with socially-aware regression for pedestrian trajectory prediction," *Pattern Recognition*, vol. 133, p. 109030, 2023.

[10] C. Wang, Y. Wang, M. Xu, and D. J. Crandall, "Stepwise goal-driven networks for trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2716–2723, 2022.

[11] Y. Chen, B. Ivanovic, and M. Pavone, "Scept: Scene-consistent, policy-based trajectory predictions for planning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 17 103–17 112.

[12] Y. Li, R. Liang, W. Wei, W. Wang, J. Zhou, and X. Li, "Temporal pyramid network with spatial-temporal attention for pedestrian trajectory prediction," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1006–1019, 2021.

[13] L. Zhou, Y. Zhao, D. Yang, and J. Liu, "Gchgat: Pedestrian trajectory prediction using group constrained hierarchical graph attention networks," *Applied Intelligence*, vol. 52, no. 10, pp. 11 434–11 447, 2022.

[14] C. Yang, H. Pan, W. Sun, and H. Gao, "Social self-attention generative adversarial networks for human trajectory prediction," *IEEE Transactions on Artificial Intelligence*, 2023.

[15] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal *et al.*, "Scene transformer: A unified architecture for predicting multiple agent trajectories," *arXiv preprint arXiv:2106.08417*, 2021.

[16] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2980–2987.

[17] H. Cheng, M. Liu, L. Chen, H. Broszio, M. Sester, and M. Y. Yang, "Gatraj: A graph-and attention-based multi-agent trajectory prediction model," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 205, pp. 163–175, 2023.

[18] W. Mao, C. Xu, Q. Zhu, S. Chen, and Y. Wang, "Leapfrog diffusion model for stochastic trajectory prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 5517–5526.

[19] I. Bae, Y.-J. Park, and H.-G. Jeon, "Singulartrajectory: Universal trajectory predictor using diffusion model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 890–17 901.

[20] A. Keysan, A. Look, E. Kosman, G. Gürsun, J. Wagner, Y. Yu, and B. Rakitsch, "Can you text what is happening? integrating pre-trained language encoders into trajectory prediction models for autonomous driving," *arXiv preprint arXiv:2309.05282*, 2023.

[21] I. Bae, J. Lee, and H.-G. Jeon, "Can language beat numerical regression? language-based multimodal trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 753–766.

[22] B. Ivanovic and M. Pavone, "Rethinking trajectory forecasting evaluation," *arXiv preprint arXiv:2107.10297*, 2021.

[23] H. Wu, T. Phong, C. Yu, P. Cai, S. Zheng, and D. Hsu, "What truly matters in trajectory prediction for autonomous driving?" *arXiv preprint arXiv:2306.15136*, 2023.

[24] S. Shridhar, Y. Ma, T. Stentz, Z. Shen, G. C. Haynes, and N. Traft, "Beelines: Motion prediction metrics for self-driving safety and comfort," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 881–887.

[25] A. Mohamed, D. Zhu, W. Vu, M. Elhoseiny, and C. Claudel, "Social-implicit: Rethinking trajectory prediction evaluation and the effectiveness of implicit maximum likelihood estimation," in *European Conference on Computer Vision*. Springer, 2022, pp. 463–479.

[26] S. Pellegrini, A. Ess, and L. Van Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*. Springer, 2010, pp. 452–465.

[27] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer graphics forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 655–664.

[28] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer, 2016, pp. 549–565.

[29] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8748–8757.

[30] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9710–9719.

[31] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[32] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 759–776.

[33] N. Uhlemann, F. Fent, and M. Lienkamp, "Evaluating pedestrian trajectory prediction methods with respect to autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[34] T. Matiisen, "Ut-adl/autoware_mini: Autoware mini is a minimalistic python-based autonomy software." 2023. [Online]. Available: https://github.com/UT-ADL/autoware_mini/

[35] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018, pp. 287–296.

[36] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[37] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[38] P. Li, H. Zhao, P. Liu, and F. Cao, "Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving," in *European Conference on Computer Vision*. Springer, 2020, pp. 644–660.

[39] J. S. Hunter, "The exponentially weighted moving average," *Journal of quality technology*, vol. 18, no. 4, pp. 203–210, 1986.

[40] X. Wang, M. A. Maleki, M. W. Azhar, and P. Trancoso, "Moving forward: A review of autonomous driving software and hardware systems," *arXiv preprint arXiv:2411.10291*, 2024.

[41] L. A. Thiede and P. P. Brahma, "Analyzing the variety loss in the context of probabilistic trajectory prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9954–9963.

[42] N. Shahroudi, M. Lepson, and M. Kull, "Evaluation of trajectory distribution predictions with energy score," in *Forty-first International Conference on Machine Learning*, 2024.