.

LiMTR: Time Series Motion Prediction for Diverse Road Users through Multimodal Feature Integration

Camiel Oerlemans[1*], **Bram Grooten**[1*], **Michiel Braat**[2], **Alaa Alassi**[2],
Emilia Silvas[1,2], **Decebal Constantin Mocanu**[3,1]
[1]Eindhoven University of Technology, [2]TNO Helmond, [3]University of Luxembourg

Abstract

Predicting the behavior of road users accurately is crucial to enable the safe operation of autonomous vehicles in urban or densely populated areas. Therefore, there has been a growing interest in time series motion prediction research, leading to significant advancements in state-of-the-art techniques in recent years. However, the potential of using LiDAR data to capture more detailed local features, such as a person's gaze or posture, remains largely unexplored. To address this, we develop a novel multimodal approach for motion prediction based on the PointNet foundation model architecture, incorporating local LiDAR features. Evaluation on the Waymo Open Dataset shows a performance improvement of 6.20% and 1.58% in minADE and mAP respectively, when integrated and compared with the previous state-of-the-art MTR. We open-source the code of our LiMTR model.[1]

1 Introduction

Time series motion prediction involves generating potential future trajectories of a road user and estimating their likelihood of occurrence based on a limited set of temporal points [24]. This is a crucial task in the safe operation of autonomous vehicles, as part of the high-level functional pipeline of autonomous vehicles: *environment perception*, *motion prediction*, *planning*, and *control* [12]. Current motion prediction solutions mainly receive a relatively coarse-grained modality; the output of the *object detection* step, which contains the objects' positions, velocities, accelerations, and bounding boxes in combination with accurate road information [24, 22]. This representation of the objects is quite efficient due to their high information density [9]. However, it potentially leaves out more fine-grained information about a target such as a person's gaze or pose direction [17, 4] during the environment perception step. Including these modalities could potentially help the model predictions for vulnerable road users, e.g. pedestrians and cyclists. Li et al. [14] have shown that light detection and ranging (LiDAR) data could effectively be used to predict pedestrian crossing. We aim to investigate how directly incorporating LiDAR data into motion prediction models can enhance the accuracy of predicted future trajectories, particularly for vulnerable road users.

We introduce a new LiDAR encoder for motion prediction called LiMTR,[2] based on the Motion Transformer by Shi et al. [21, MTR]. We provide the encoder with LiDAR data pertaining only to the target road user, to help the model focus on features such as a person's pose or gaze direction. We refer to this as providing *local* LiDAR features, compared to global scene-level features. We demonstrate its effectiveness by experimentation on the Waymo Open Dataset [4, WOD, see Appendix A], gaining an overall performance improvement of 6.20% and 1.58% in minimum average displacement error (minADE) and mean average precision (mAP) respectively, compared to the baseline without LiDAR.

[*]Equal contribution. Corresponding author: b.grooten@tue.nl
[1]See https://github.com/Cing2/LiMTR
[2]For LiDAR Motion Transformer, pronounced as *Limiter*.

NeurIPS 2024 Workshop on Time Series in the Age of Large Models.

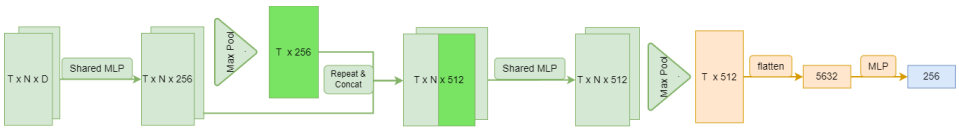

Figure 1: Our LiDAR encoder. Point compression in green, time in orange, with the final feature in blue. The variables represent timesteps (T), points (N), and feature dimension per point (D).

2 Time Series Motion Prediction

In this section, we explain the task of time series motion prediction, specifically how it is defined for the Waymo Motion Prediction challenge. For this task, the model is provided with 1 second of past information on a scene at 10Hz, and needs to generate up to m potential future trajectories for each of the n road users, referred to as agents, where $m = 6$ and $1 \leq n \leq 8$ for a single scene. A trajectory consists of the (x, y) coordinates on a local map for 8 seconds into the future at 2Hz. For further background on motion prediction, refer to Appendix B.

2.1 Performance Metrics

We compare the models with metrics used for the Waymo Motion Prediction challenge, also commonly employed in research; for full definitions see [7]. All metrics are computed at time horizons of 3, 5, and 8 seconds and for each class of road user (pedestrian, cyclist, vehicle). Lastly, the mean over these time horizons and classes is taken to compute the final metric.

minADE$\downarrow \in [0, \infty)$ Minimum Average Displacement Error, takes the minimum of the L2-norm between the ground truth trajectory and the m predicted trajectories.

MR$\downarrow \in [0, 1]$ Miss rate is the fraction of trajectory sets where none of the trajectories are correct. A trajectory is determined *correct* if its final position is within a certain threshold distance of the ground truth. This distance threshold is different for lateral and longitudinal distances and scales with the velocity of the target at time 0.

mAP$\uparrow \in [0, 1]$ Mean Average Precision uses the same definition of correctness as the miss rate to determine true positives and false positives, and only one true positive is allowed per target. The predicted trajectories per target are bucketed over 8 behaviors, i.e. left turn, right turn, etc. Then the area under the precision-recall curve is calculated across various thresholds to get the average precision per bucket. The mean value of these buckets is the mAP.

3 LiMTR

We proceed to design our LiDAR encoder for motion prediction. We provide our network with local sections of the raw point cloud directly, training all of our components end-to-end.

3.1 Local LiDAR Features

The input to our LiDAR encoder is the subset of 3D LiDAR points of each road user we are currently predicting. We select all 3D LiDAR points that fall into the provided bounding box of the object, where we include a 15% margin, to include possible missed points due to errors in bounding box labeling. We use this subset to greatly reduce computational requirements, and we hypothesize that the LiDAR points of the target are most informative for its future trajectory. This may include potential features such as a person's gaze direction or a cyclist signaling to change course by pointing.

The 3D LiDAR points are centered on the target and rotated to align with the forward-facing direction of the target, similar to the preprocessing of the trajectories [3, 21]. The number of LiDAR points taken is limited to 512; if the number of available points exceeds this, a random subset is taken, if fewer are available the set is padded with zeros. The PointNet architecture [18] has been shown to be quite robust to random removal of points. Furthermore, we add a one-hot encoding of the target class to each LiDAR point, similar to how Vora et al. [25] add the probabilities of another detector. Additionally, we include the intensity feature and leave out range and elongation, see subsection 4.4.

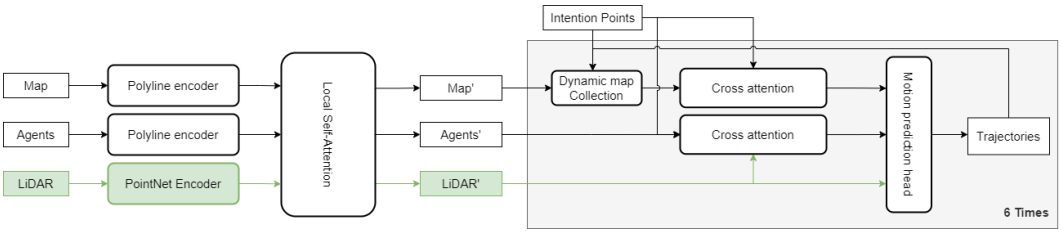

Figure 2: LiMTR architecture based on the MTR [21] model, with our LiDAR encoder in green.

3.2 LiDAR Encoder

Figure 1 presents the architecture of our LiDAR encoder, based on the PointNet [18] architecture, which was chosen as it can directly process LiDAR points without the need for voxelization or other extensive preprocessing. The diagram shows the dimensions of how a single-object LiDAR point set is transformed into a feature vector; $T = 11$ for the number of frames and $N = 512$ for the number of points. The encoder consists of two parts: the first part compresses the point dimension and the second compresses the time dimension, resulting in a single feature vector of 256, highlighted in green, orange, and blue, respectively. The point compression part uses shared MLPs over the feature dimension and a max-pool layer. Max-pooling is permutation invariant, which is advantageous as the LiDAR points do not have a distinct order [18]. Additionally, as done in [18, 21], we add a global feature by taking the max-pool, repeating the feature, and concatenating it to the feature dimension. The time compression part flattens the time dimension and then applies multiple MLP layers.

The three MLP blocks consist of 12 linear layers, each followed by Batch Normalization [13] and ReLU activation [11]. The hidden dimensions within each MLP block are $[256, 512, 1024]$. From scaling experiments, discussed in subsection 4.3, we determined that 12 layers for each shared MLP gave optimal performance. Figure 2 shows how the LiDAR encoder is integrated into MTR [21]. The LiDAR data is first passed through the encoder and then sent to the local self-attention module. Next, the LiDAR feature is combined with the processed agents' features to be used in cross-attention and is also directly sent to the motion prediction head.

4 Results and Discussion

4.1 Experimental setup

We train LiMTR with the AdamW [15] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ a learning rate of $3 \cdot 10^{-4}$, and 0.01 weight decay. We use a Linear Decay learning rate scheduler with a linear warmup period of 5%, following [5]. We train the model for 60 epochs and a batch size of 192 scenarios on 12 NVIDIA A100 GPUs, which takes approximately 80 hours. We compare our LiMTR model against multiple baselines, the most important one being MTR [21] without the LiDAR modality. Furthermore, we compare against MGTR [8] which uses global scene-level features, and against Wayformer [4], that uses a pretrained object detection model for LiDAR.

4.2 Main Results

Table 1 presents our results on the Waymo Motion Prediction [4] validation set. The bottom section shows the performance of Wayformer with [4] and without [17] LiDAR, for which the authors only reported at $t = 8$s. The top section shows averages over $t = 3, 5, 8$s, as is standard practice [8, 21]. Our LiDAR approach shows consistent performance improvement on all metrics compared to MTR with an average increase of 6.20% and 1.58% in minADE and mAP, respectively. In comparison to Wayformer with a pretrained LiDAR model, LiMTR achieves an improved mAP, but a slightly higher minADE. Their LiDAR solution predicts all objects in the scene which might explain the extra benefit on that metric. LiMTR performs especially well on vulnerable road users, such as pedestrians and cyclists. These show the largest increase in mAP, suggesting that gazes or poses may indeed be inferred from the LiDAR data.

Table 1: Performance on the Waymo Open Dataset [4]. The **top** section results are averaged over $t = 3$, 5, and 8 seconds, the **bottom** section are reported at $t = 8$s. Results of Wayformer, Wayformer+LiDAR, and MGTR are taken from [17, 4, 8]. For MGTR [8], no minADE or MR metrics are reported. Best results per section are in shown **bold**.

	Avg		Vehicle			Pedestrian			Cyclist		
Model	mAP↑	minADE↓	mAP↑	minADE↓	MR↓	mAP↑	minADE↓	MR↓	mAP↑	minADE↓	MR↓
MGTR [8]	**0.45**		**0.46**			**0.47**			**0.40**		
MTR [21]	0.4029	0.6812	0.4165	0.8845	0.1756	0.4292	0.3693	0.0855	0.3630	0.7896	0.2097
LiMTR (Ours)	0.4093	**0.6389**	0.4214	**0.7990**	**0.1686**	0.4363	**0.3613**	**0.0810**	0.3702	**0.7565**	**0.2041**
Wayformer [17]	0.33	0.91	0.35	1.10	0.18	0.35	**0.54**	0.11	0.29	1.08	0.22
Wayformer+LiDAR [4]	0.34	**0.90**	**0.37**	**1.09**	**0.17**	0.37	**0.54**	**0.10**	0.28	**1.06**	**0.21**
MTR [21]	0.3459	1.1313	0.3426	1.5151	0.2254	0.3826	0.5985	0.1093	0.3125	1.2802	0.2235
LiMTR (Ours)	**0.3508**	1.0496	0.3458	1.3640	0.2156	**0.3894**	0.5838	**0.1002**	**0.3171**	1.2009	**0.2148**

4.3 Scaling experiments

We conduct a network scaling experiment to investigate the LiDAR encoder size required to extract relevant features. We increase the depth of the 3 MLP blocks simultaneously from 2 to 14 layers with a step of 2. The resulting model ranges from 7.8M to 24M parameters. In the scaling and ablation experiments, we train for 30 epochs on 10% of the data, using batch size 64 and learning rate 10^{-4}. Figure 3 shows the LiDAR encoder size in the number of parameters to minADE. The plot indicates that model performance correlates with size, where larger models are preferred. For the main model, we use the 12-layer, 22M parameter model as it demonstrates better performance on the mAP metric.

4.4 Ablation study

We perform ablation experiments to determine two aspects: which LiDAR input features are important and what is the effect of using multiple LiDAR frames. First, as described in Appendix A, the dataset also includes additional LiDAR features; range, intensity, and elongation. We trained our model with each feature alone and with all three included. Second, over the 1 second of past data, we have 11 timeframes of LiDAR. To determine the importance of the time series aspect of the LiDAR encoder, we trained LiMTR with 11, 6, 3, or 1 LiDAR timeframe(s) provided. Each option was trained with 2 different seeds and the mean and standard deviation of the minADE and mAP are shown in Table 2.

Time series. For the number of timestamps, there is no clear distinction on which is better. The full 11 timestamps displays the best performance on mAP, however, the worst performance on minADE. The variance between the runs is too large to determine a clear choice.

Features. The intensity feature performs better than range and elongation when looking at the minADE metric. Further, the experiments with all features included show similar performance to the intensity feature. This suggests that the intensity feature meaningfully contributes to the task.

Table 2: Ablation study on the number of time steps and LiDAR point features.

Time steps	mAP ↑	minADE ↓	LiDAR features	mAP ↑	minADE ↓
1	0.3333 (0.0005)	0.9428 (0.0097)	Range	0.3336 (0.0036)	0.9608 (0.0001)
3	0.3311 (0.0019)	0.9376 (0.0135)	Intensity	0.3343 (0.0037)	**0.9414** (0.0041)
6	0.3333 (0.0031)	**0.9337** (0.0110)	Elongation	0.3318 (0.0052)	0.9734 (0.0066)
11	**0.3388** (0.0003)	0.9507 (0.0166)	All	**0.3388** (0.0003)	0.9507 (0.0166)

5 Conclusion

We propose LiMTR, a novel time series motion prediction model for autonomous driving, that incorporates local LiDAR features. The model is trained directly on LiDAR data allowing it to capture intricate motion details. When compared to the previous state-of-the-art MTR model, we show that LiMTR gains 6.20% and 1.58% in minADE and mAP, respectively. By only providing our model with the LiDAR data of a target road user, we help it focus on target-specific features. This makes LiMTR complementary to other techniques that incorporate global LiDAR features.

Acknowledgments and Disclosure of Funding

This work is part of the AMADeuS project of the Open Technology Programme (project number 18489), which is partly financed by the Dutch Research Council (NWO). This research used the Dutch national e-infrastructure with the support of the SURF Cooperative, using grant number EINF-6221.

References

[1] 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. *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 11618–11628, 2020. (Cited on page 7)

[2] S. Casas, W. Luo, and R. Urtasun. IntentNet: Learning to Predict Intention from Raw Sensor Data. In *Mach. Learn. Res.*, pages 947–956. PMLR, 2018. (Cited on page 7)

[3] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov. MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction. In *Mach. Learn. Res.*, pages 86–99. PMLR, 2019. (Cited on page 2, 8)

[4] K. Chen, R. Ge, H. Qiu, R. Ai-Rfou, C. R. Qi, X. Zhou, Z. Yang, S. Ettinger, P. Sun, Z. Leng, M. Mustafa, I. Bogun, W. Wang, M. Tan, and D. Anguelov. WOMD-LiDAR: Raw Sensor Dataset Benchmark for Motion Forecasting. *arXiv2304.03834*, 2023. (Cited on page 1, 3, 4, 7)

[5] A. Defazio, A. Cutkosky, H. Mehta, and K. Mishchenko. When, Why and How Much? Adaptive Learning Rate Scheduling by Refinement. *arXiv2310.07831*, 2023. (Cited on page 3)

[6] N. Djuric, H. Cui, Z. Su, S. Wu, H. Wang, F. C. Chou, L. S. Martin, S. Feng, R. Hu, Y. Xu, A. Dayan, S. Zhang, B. C. Becker, G. P. Meyer, C. Vallespi-Gonzalez, and C. K. Wellington. MultiXNet: Multiclass multistage multimodal motion prediction. In *IEEE Intell. Veh. Symp.*, pages 435–442. IEEE Press, 2021. (Cited on page 7)

[7] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov. Large Scale Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset. *IEEE Int. Conf. Comput. Vis.*, pages 9690–9699, 2021. (Cited on page 2, 7)

[8] Y. Gan, H. Xiao, Y. Zhao, E. Zhang, Z. Huang, X. Ye, and L. Ge. Multi-Granular Transformer for Motion Prediction with LiDAR. *arXiv2312.02409v1*, 2023. (Cited on page 3, 4, 7)

[9] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid. VectorNet: Encoding HD maps and agent dynamics from vectorized representation. *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2020. (Cited on page 1, 8)

[10] J. Gu, C. Sun, and H. Zhao. DenseTNT: End-to-end Trajectory Prediction from Dense Goal Sets. In *IEEE Int. Conf. Comput. Vis.*, pages 15283–15292, 2021. (Cited on page 8)

[11] R. H. Hahnioser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. Digital selection and analogue amplification coexist in a cortex- inspired silicon circuit. *Nature*, 405(6789):947–951, 2000. (Cited on page 3)

[12] R. Huang, G. Zhuo, L. Xiong, S. Lu, and W. Tian. A Review of Deep Learning-Based Vehicle Motion Prediction for Autonomous Driving. *Sustainability*, 15(20), 2023. (Cited on page 1)

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Int. Conf. Mach. Learn.*, pages 448–456, 2015. (Cited on page 3)

[14] J. Li, X. Shi, F. Chen, J. Stroud, Z. Zhang, T. Lan, J. Mao, J. Kang, K. S. Refaat, W. Yang, E. Ie, and C. Li. Pedestrian Crossing Action Recognition and Trajectory Prediction with 3D Human Keypoints. *IEEE Int. Conf. Robot. Autom.*, pages 1463–1470, 2023. (Cited on page 1)

[15] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *Int. Conf. Learn. Represent. ICLR*, 2019. (Cited on page 3)

[16] W. Luo, B. Yang, and R. Urtasun. Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net. In *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 3569–3577, 2018. (Cited on page 7)

[17] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp. Wayformer: Motion Forecasting via Simple & Efficient Attention Networks. In *IEEE Int. Conf. Robot. Autom.*, volume 2023-May, pages 2980–2987, 2023. (Cited on page 1, 3, 4, 7, 8)

[18] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. *30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR*, pages 77–85, 2017. (Cited on page 2, 3, 8)

[19] M. M. Sánchez, E. Silvas, J. Elfring, and R. van de Molengraft. Robustness benchmark of road user trajectory prediction models for automated driving. *IFAC-PapersOnLine*, 56(2):4865–4870, 2023. (Cited on page 7)

[20] M. M. Sánchez, E. Silvas, D. Pogosov, and D. C. Mocanu. A Hybrid Framework Combining Vehicle System Knowledge with Machine Learning Methods for Improved Highway Trajectory Prediction. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 444–450. IEEE, 2020. (Cited on page 7)

[21] S. Shi, L. Jiang, D. Dai, and B. Schiele. Motion Transformer with Global Intention Localization and Local Movement Refinement. *Adv. Neur. Inf. Process. Sys.*, 2022. (Cited on page 1, 2, 3, 4, 7, 8)

[22] S. Shi, L. Jiang, D. Dai, and B. Schiele. MTR++: Multi-Agent Motion Prediction with Symmetric Scene Modeling and Guided Intention Querying. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024. (Cited on page 1, 8)

[23] P. Sun, M. Tan, W. Wang, C. Liu, F. Xia, Z. Leng, and D. Anguelov. SWFormer: Sparse Window Transformer for 3D Object Detection in Point Clouds. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, pages 426–442, 2022. (Cited on page 8)

[24] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov, and B. Sapp. MultiPath++: Efficient Information Fusion and Trajectory Aggregation for Behavior Prediction. *IEEE Int. Conf. Robot. Autom.*, 2022. (Cited on page 1, 8)

[25] S. Vora, A. H. Lang, B. Helou, and O. Beijbom. Pointpainting: Sequential fusion for 3D object detection. In *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 4603–4611, 2020. (Cited on page 2)

[26] D. Ye, Z. Zhou, W. Chen, Y. Xie, Y. Wang, P. Wang, and H. Foroosh. LidarMultiNet: Towards a Unified Multi-Task Network for LiDAR Perception. *37th AAAI Conf. Artif. Intell. AAAI 2023*, 37(3):3231–3240, 2023. (Cited on page 7)

Appendix

A Waymo Open Dataset

The release of the LiDAR data for the Waymo Open Dataset (WOD) [4] opened up new possibilities for using LiDAR data for motion prediction, which has remained low due to the low availability of large motion prediction datasets with LiDAR data. The Waymo Motion Prediction dataset is two orders of magnitude larger compared to the second largest open motion prediction dataset with LiDAR NuScenes [1] with 1k scenarios compared to 104k scenarios for Waymo [4]. The scenarios in the WOD are often interesting traffic situations, such as an intersection, and the data includes tracked and labeled road users in combination with road layout information. Each scene is 9 seconds long and recorded at 10 Hz. The data for a scenario consists of three parts: the road information, the agents' information, and LiDAR data. The road and agent information is made with their offboard perception system from the raw sensor data of the car [7].

Road information. The road information consists of static map features and dynamic features. The latter are traffic light states connected to a lane, given at each timestamp. The static map features represent the 2D road layout and contains a polyline or outline of the following types: lane, road edge, road line, stop sign, crosswalk, speedbump, and driveway.

Agents information. The agents' information consists of information on the dynamic objects of the scenario, which are classified as vehicles, cyclists, or pedestrians. Each object consists of a state for every timestamp, where the state has a position in local x, y, and z coordinates, velocity in x and y, and the object's bounding box.

LiDAR. The LiDAR data is collected from five LiDAR sensors, one on top of the vehicle with a sensor resolution of 64 pixels in height and 2650 pixels in width, and four on the sides of the vehicle with a sensor resolution of 116x150. Each pixel records its position in x, y, and z and its corresponding features range, intensity, and elongation.

B Motion Prediction Background

In this Appendix, we describe related machine learning work on the problem of time series motion prediction using LiDAR data and explain the MTR [21] model used as a baseline for this work. The field of time series motion or trajectory prediction is activity searching for the best and safest algorithms to use for autonomous driving [4, 8, 17, 19, 20]. The starting point for this research is the Waymo Motion Prediction challenge. For this task, the model is given the first second of past data and should predict the exact locations of various road users up to 8 seconds into the future. The provided data consists of road information, road users' information, and raw LiDAR data.

B.1 Motion Prediction with LiDAR

There are a few motion prediction works that make use of LiDAR data. Early work by Uber includes LiDAR by voxelizing the data into a birds-eye view (BEV) 3D grid with a resolution of 0.2 [16, 2] or 0.16 [6] meters. The voxelized LiDAR is then processed with convolutional neural network (CNN) backbones to extract relevant features and then jointly perform object detection and motion prediction. Similarly, MGTR [8] also includes BEV 3D voxelized LiDAR data, with two different resolutions of 0.8 and 1.6 meters. These are encoded with a pre-trained LiDARMultiNext [26] model and used in their Transformer encoder-decoder architecture to do motion prediction. The MGTR model achieved state-of-the-art performance winning the Waymo Motion Prediction challenge in 2023. The voxelization is an important processing step to reduce computation requirements and makes it easier to process with CNN backbones which are good at capturing local structures in the LiDAR data [2]. However, the voxelization might leave out finer details due to lowering the resolution, where details that require a higher resolution will not be distinguishable anymore, such as potentially a person's pose or gaze direction.

Waymo, with their release of the 2023 Waymo Open Dataset [4], experimented with adding LiDAR data to motion prediction models. They use the final embeddings and detected boxes from a LiDAR

3D object detection model, SWFormer [23], and include these in the WayFormer [17] model. They lowered the prediction threshold of the object detection model allowing it to detect more objects. Wayformer with the addition of the LiDAR data gained an increase in mAP from 0.35 to 0.37 for both vehicles and pedestrians. However, there was no improvement for cyclists, which may be due to the features their LiDAR encoder retrieves for bounding box detection might leave out information relevant to motion prediction. This suggests a need to investigate if the features extracted by an object detection model are adequate for motion prediction.

B.2 Motion TRansformer: MTR

The MTR model [21] is the baseline for this work and the model we extend with our LiDAR approach. We select MTR as the open-source state-of-the-art during our research, following the first edition of the MTR model [21]. Shi et al. later published the improved MTR++ [22] version that includes symmetric scene context encoding and joint motion decoder, allowing the model to predict multiple objects in the same scene more efficiently. The MTR model uses the Transformer encoder-decoder architecture. Figure 2 shows an outline of the model, where the grey parts are the original architecture.

Data representation. The road information and agent history are encoded using the vectorized representation [9]. The coordinates of the road and agent polylines are centered and rotated on the current target agent coordinate system, adopting an agent-centric strategy [10, 24].

Context encoder. First, a polyline encoder based on the PointNet [18] architecture is used to encode the agent and road vectors to produce their respective features. These encoded features are then processed with local self-attention by only attending to nearby objects. This is done by applying k-nearest neighbors (KNN) on each object based on its initial position to collect a neighborhood of k neighbors. This reduces the computation requirements and crucially preserves the local structure of the scene [22].

Motion decoder. The motion decoder uses the transformer-based decoder structure to apply cross-attention on the previously encoded road and agent features. They employ learnable query embeddings from intention points to act as intention queries. The intention points are made by applying k-means clustering on the set of relative final locations of all targets in the training set, separately for the three classes, vehicles, pedestrians, and cyclists. These intention points act as anchor points, similar to [10] goal points, and are crucial to generating a distinct set of trajectories. The query embeddings are then used in cross-attention with the agent features to collect relevant features for each intention point.

Before applying attention to map features, a *dynamic map collection* technique is used. Similar to the context encoder local attention method, this also applies KNN on the map features to collect a set of relevant features. For the first decoder layer, the distance to the intention point is used for KNN, and in consecutive layers, the smallest distance to any point in the previously predicted trajectories. This allows the model to iteratively refine the trajectory on relevant features.

The motion prediction head takes the queried features per intention point and uses multilayer perceptron (MLP) layers to generate the trajectories with accompanying probabilities. A trajectory consists of 80 timestamps which cover 8 seconds with 10Hz. Following [3, 24] they represent the trajectories as Gaussian Mixture Model (GMM) components x, y, std_x, std_y, ρ, for each timestamp. This is an important part of modeling the ambiguous behavior of the agents.

Loss. The loss function consists of three parts, the negative log-likelihood loss for the GMM components, an $L1$ loss on the predicted velocity, and a cross-entropy loss on the probabilities of the trajectories.

C Analysis

In this section we include additional figures and information that complement the scaling and ablation studies in the paper. For the time series ablation: we take equally spaced timestamps over the 11 total. Thus, e.g., for 6 time steps these are the frames $(0, 2, 4, 6, 8, 10)$, and for the 1 timeframe we use the last time step.

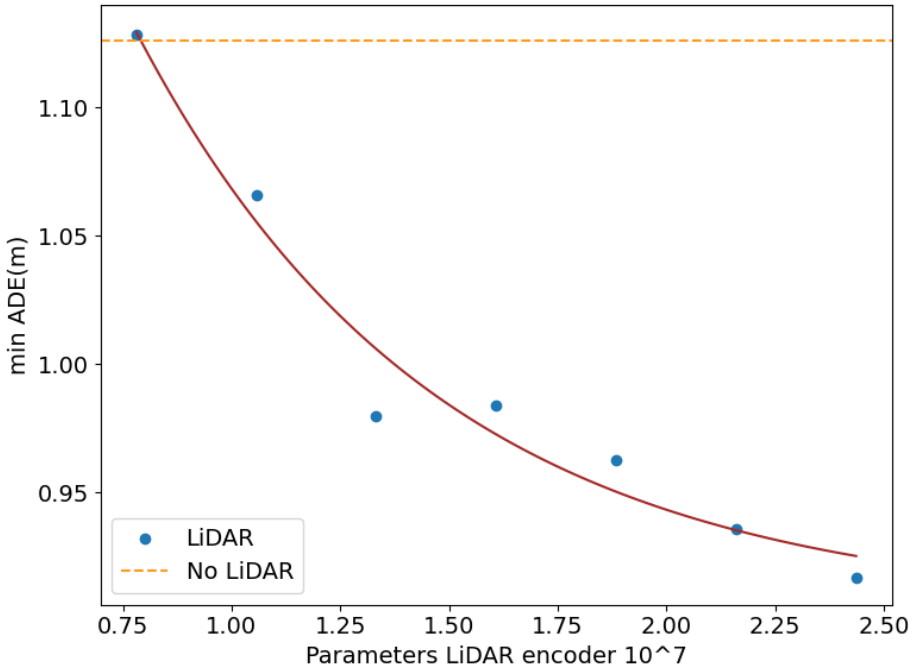

Figure 3: Scaling of the LiDAR encoder network, showing the number of parameters to model performance in minADE, including a horizontal line denoting baseline MTR performance without LiDAR, and a fitted exponential function.

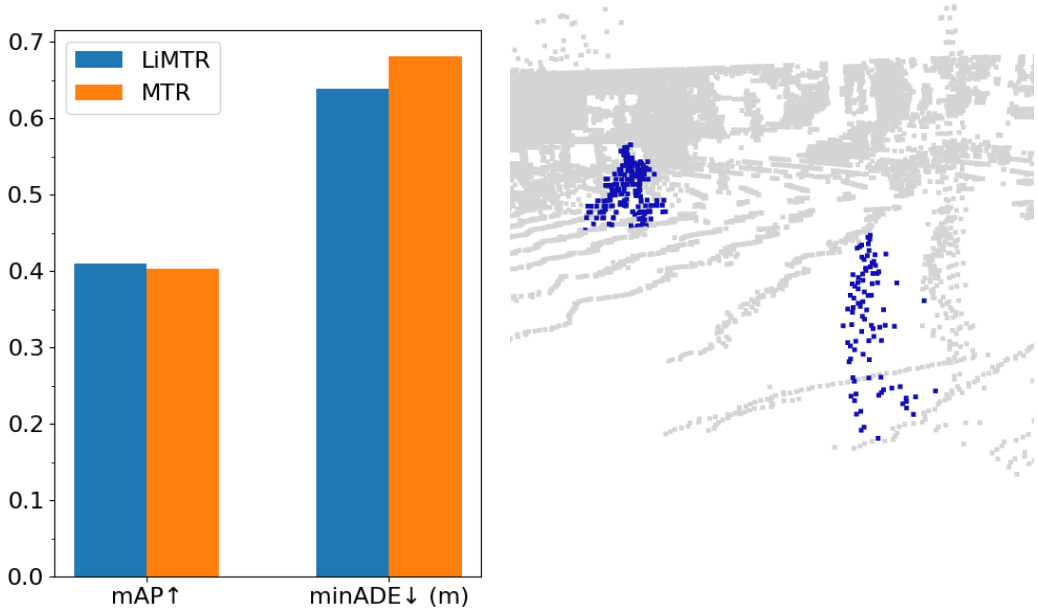

Figure 4: (**Left**) Performance of LiMTR (ours) and MTR. (**Right**) Local LiDAR data of a cyclist and a pedestrian in a scene in blue. Our LiMTR model receives time series LiDAR data at 10Hz, capturing the movement of such point clouds per target object.