Motion Prediction: Comparative Study & Introducing a New Model

Syed Ali Haider[1]
Courant Institute of Mathematical Sciences
New York University, NY, USA, 10003
sh6070@nyu.edu

Harsh Tambi[1]
Courant Institute of Mathematical Sciences
New York University, NY, USA, 10003
ht2279@nyu.edu

Abstract—**Our research presents a novel model for predicting steering angles in autonomous vehicles, leveraging insights from the DAVE-2 system and incorporating elements inspired by the Motion Transformer (MTR) and Multi-Granular Transformer (MGTR) methodologies. The proposed architecture encompasses an input layer for receiving resized images, a reshape layer for reformatting input data, and a positional encoding layer for spatial awareness. The core comprises Transformer Encoder blocks stacked to learn intricate features crucial for steering angle prediction. A dense output layer maps these features to predicted steering angles, facilitating vehicle control. Despite resource constraints, our model achieves a promising accuracy of 95.5% on the test dataset. This research contributes to advancing motion prediction in autonomous driving technology, with implications for enhancing vehicle safety and efficiency.**

I. Introduction

The advent of autonomous driving technology heralds a transformative era in transportation, promising safer and more efficient journeys on our roadways. At the heart of this technological revolution lies the critical ability to accurately predict motion, particularly in dynamic and interactive environments. Motion prediction serves as the cornerstone for enabling autonomous vehicles to navigate complex scenarios such as merges, unprotected turns, and interactions with other road users. However, achieving precise motion forecasting poses formidable challenges, given the multitude of factors influencing vehicle trajectories and the dynamic nature of traffic scenarios.

In recent years, researchers have explored diverse methodologies to address the challenge of motion prediction in autonomous vehicles. These methodologies span from conventional approaches relying on convolutional neural networks (CNNs) Bojarski et al. (2016) to cutting-edge techniques harnessing the capabilities of transformer-based architectures. Noteworthy among these advancements are the Motion Transformer methodologies, which have showcased remarkable capabilities in capturing intricate motion patterns and contextual information.

Inspired by these advancements, our research endeavors to develop a novel model for steering angle prediction in autonomous vehicles. Drawing insights from the DAVE-2 Bojarski et al. system, which utilizes camera data to infer steering commands, and integrating principles from Shi et al. (2023) and Gan et al. (2024) methodologies, our model aims to enhance the accuracy and robustness of motion prediction in dynamic driving scenarios.

In this paper, we present a comprehensive exploration of our proposed model, elucidating its architecture, key components, and performance metrics. Through experimentation and comparative analysis with existing methodologies, we assess the efficacy of our model in advancing the state-of-the-art in autonomous driving technology.

II. Related Work

Motion prediction, especially in the context of self-driving cars, has gained significant attention in recent years. It revolves around forecasting the trajectories of objects such as vehicles based on road layouts and their past motion patterns. Early approaches transformed road maps and object trajectories into image representations, leveraging convolutional neural networks for analysis Park et al. (2020); Marchetti et al. (2020). Another prominent method is VectorNet Gao et al. (2020), which represents roads and object paths as polylines, renowned for its computational efficiency.

Various strategies have been explored to predict multiple potential future paths for objects. Some researchers employed Gaussian models for path estimation Alahi et al. (2016). IntentNet Casas et al. (2021), instead of predicting exact trajectories, focused on foreseeing the intentions or actions of objects . Goal-based methods like DenseTNT Zhao et al. (2021) inferred the probable destinations of objects and extrapolated their complete trajectories from those points . However, these recent methods, while aiming to predict long-term movements, encountered certain limitations. IntentNet Casas et al. (2021), for instance, faced trade-offs wherein scaling to a large number of agents led to excessive computational and memory requirements, while reducing agent counts compromised model performance. Additionally, Zhao et al. reported slow convergence rates. Addressing these challenges, Shi et al. proposed a solution using a small set of ”motion query pairs” to predict specific movement patterns more efficiently.

III. Motion Transformer (MTR) 2022

In their work, the team behind MTR Shi et al. (2023) brings to light the powerful potential of Transformer architectures in motion prediction. Inspired by the remarkable success of

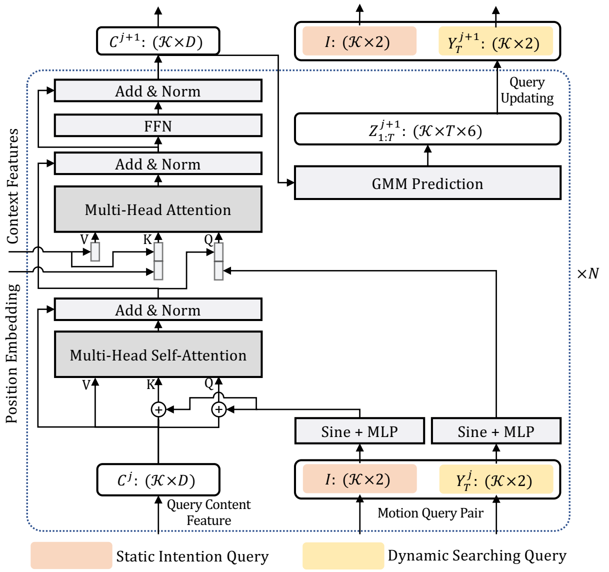


Fig. 1: he network structure of our motion decoder network with motion query pair as shown in Shi et al. (2023)

Transformers across various domains such as language understanding and image analysis Vaswani et al. (2017) Inspired from the success of models like DETR Carion et al. (2020), which leverage Transformer-like structures for object detection tasks, the authors tailor this approach to predicting object movements. Central to their methodology is the introduction of ”motion query pairs,” a concept designed to facilitate nuanced understanding and prediction of object trajectories.

The Motion Transformer (MTR) model comprises three key components: trajectory prediction, road context aggregation, and prediction refinement using a Transformer model.

A. *Encoding Scene Context*

Understanding the contextual intricacies of a scene is pivotal for accurate movement prediction. While existing methodologies employ diverse strategies, MTR focuses on preserving the locality structure of the scene. To achieve this, MTR employs a Transformer encoder network with local self-attention mechanisms. This encoder processes input trajectories and road maps, encoding them into feature representations while preserving the crucial local structure through local self-attention mechanisms.

B. *Localized Attention Mechanisms*

A distinguishing feature of MTR lies in its localized attention mechanisms, crucial for preserving the spatial relationships within the scene. The attention module within each Transformer encoder layer is crafted to maintain the locality structure. Specifically, the attention module is formulated as:

$$\begin{aligned} G^j = \text{MultiHeadAttn}\big(&\text{query} = G^{j-1} + \text{PE} * G^{j-1}, \\ &\text{key} = \kappa(G^{j-1}) + \text{PE} * \kappa(G^{j-1}), \\ &\text{value} = \kappa(G^{j-1})\big) \end{aligned}$$

Here, G^j represents the output of the attention mechanism at layer j of the Transformer encoder. The **MultiHeadAttn** function enables simultaneous attention computation across multiple heads, capturing diverse patterns in the input. The **query** component encapsulates the current input, while the **key** and **value** components provide contextual information. **Sinusoidal Position Encoding** injects positional information into the input embeddings, aiding in spatial understanding.

The function $\kappa(\cdot)$ identifies the k closest polylines to each query polyline, facilitating the preservation of local scene structure. This ensures that the model attends to relevant spatial features, leading to contextually aware predictions.

C. *Trajectory Prediction and Refinement*

MTR adopts a comprehensive approach to trajectory prediction and refinement. Future trajectories and velocities of all agents are densely predicted using a regression head on A_{past}:

$$S_{1:T} = \text{MLP}(A_{\text{past}})$$

Here, S_i encapsulates the future position and velocity of each agent at time step i, with T representing the number of future frames to predict. Subsequently, the future states of agents are encoded as features A_{future} using the same polyline encoder. These features are then concatenated with A_{past} and processed by MLP layers to enrich the feature representation A.

The decoder network further refines the predicted trajectories using ”motion query pairs,” each comprising static intention queries and dynamic searching queries. These queries aid in localizing potential motion intentions and refining local movement information, respectively.

D. *Multimodal Motion Prediction with Gaussian Mixture Model*

Recognizing the multimodal nature of agent behaviors, MTR employs a Gaussian Mixture Model (GMM) to predict multimodal distributions of future trajectories. Each decoder layer appends a prediction head to generate future trajectories, predicting parameters such as means (μ_x, μ_y), standard deviations (σ_x, σ_y), correlation coefficients (ρ), and probabilities (p) for each Gaussian component. This framework enables the generation of multimodal predictions, effectively capturing the uncertainty and diversity of agent motion in dynamic scenes.

E. *Training Losses*

The training process of the MTR model involves optimizing two distinct loss functions:

1) Auxiliary $L1$ Regression Loss:: This loss function minimizes the absolute differences between predicted and ground-truth values, facilitating precise trajectory prediction.

2) Gaussian Regression Loss: : Utilizing a negative log-likelihood loss based on the Gaussian mixture model, this loss maximizes the likelihood of the ground truth trajectory, ensuring robust trajectory prediction across diverse scenarios.

IV. Multi-Granular TRansformer (MGTR) 2023

In this paper, the introduced Multi-Granular Transformers (MGTR) Gan et al. (2024) framework addresses challenges of building autonomous vehicles by employing a Transformer-based encoder-decoder network that integrates context features at multiple granularities. This allows for a more comprehensive understanding of the environment and the diverse behaviors of traffic agents. A key innovation of MGTR is its utilization of LiDAR point cloud data, which provides dense 3D context information for motion prediction. By incorporating LiDAR semantic features through an off-the-shelf extractor, MGTR enhances its ability to perceive the environment accurately. Additionally, the framework introduces a motion-aware context search mechanism, improving both accuracy and efficiency in predicting agent motions. Through its Transformer architecture, MGTR processes multimodal inputs including agent history states, map elements, and LiDAR embeddings. These inputs are encoded into sets of tokens at various granular levels, enabling the model to capture fine details and nuanced relationships within the scene. By iteratively refining predictions within the decoder using a Gaussian Mixture Model (GMM), MGTR achieves state-of-the-art performance on the Waymo Open Dataset motion prediction benchmark, demonstrating its effectiveness in handling complex autonomous driving scenarios.

A. *Multimodal Multi-Granular Inputs:*

Agent and map representation: Agents' historical states are encoded into vectorized polylines, capturing information like position, velocity, heading angle, etc. Map elements, such as road centerlines, are sampled at different granularities to accommodate various movement ranges. LiDAR integration: LiDAR data, providing rich 3D context, is incorporated using voxel features extracted by a pre-trained segmentation network. LiDAR context features are obtained at multiple granularities through average pooling. Motion-aware context search: To manage computational complexity, a context search mechanism selects relevant map and LiDAR tokens based on an agent's current velocity, ensuring efficient feature learning and encoding of meaningful context.

B. *Transformer Encoder:*

Token aggregation and encoding: Multi-granular tokens undergo refinement through layers of Transformer encoder, employing self-attention and feed-forward networks. Local attention mechanisms are utilized for better capturing neighboring information. Future state enhancement: Future trajectories of agents are predicted and encoded, considering both historical trajectories and potential future movements, thus enriching the agent features fed into the decoder.

C. *Transformer Decoder:*

Intention goal set: Representative intention goals are generated using clustering algorithms on ground truth trajectory endpoints, enabling the model to capture implicit motion modes. Token aggregation with intention goal set: Features from the encoder are aggregated using self-attention and cross-attention mechanisms, incorporating trajectory-aware and motion-aware context searches. Multimodal motion prediction with GMM: Future trajectories are modeled using Gaussian Mixture Models (GMM), with classification and regression heads predicting trajectory modes and parameters.

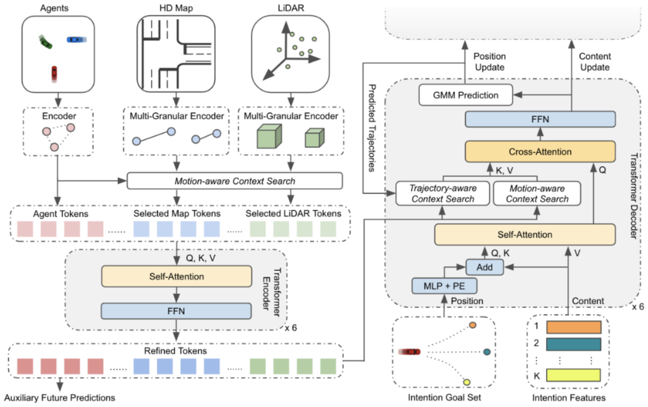


Fig. 2: MGTR Gan et al. (2024) Architecture

D. *Training Loss:*

The training loss combines auxiliary task loss, classification loss, and GMM loss. Auxiliary task loss ensures alignment between predicted and ground truth trajectories. Classification loss measures the accuracy of predicted intention probabilities. GMM loss evaluates the likelihood of predicted trajectories, using a hard-assignment strategy to specialize each mode for distinct agent behaviors.

E. *Contributions:*

Integration of multimodal inputs: MGTR seamlessly incorporates information from agents' historical states, high-definition maps, and LiDAR data, enabling a comprehensive understanding of the driving environment. Multi-granular representation: By representing inputs at various granularities, the model captures both macroscopic and detailed spatial information, enhancing prediction accuracy. Efficient context search: Motion-aware context search optimizes computational resources by selecting relevant tokens for feature learning, improving efficiency without sacrificing performance. End-to-end motion prediction: The model provides a holistic solution for motion prediction, from encoding diverse inputs to generating multimodal future trajectories, facilitating safe and efficient autonomous driving

V. DAVE-2

The DAVE-2 system, as outlined in Bojarski et al. (2016), employs three windshield-mounted cameras to capture timestamped video and steering angle data from the driver. Steering commands are represented as 1/r to ensure uniformity across car geometries, facilitating smooth transitions between turns. Training data includes single images paired with corresponding steering commands, augmented with off-center and rotated images to refine error correction. Although viewpoint transformations may introduce distortions for objects above ground level, they have negligible impact on network training. Training involves feeding images into a CNN, with adjustments made via backpropagation to minimize differences between computed and desired steering commands. Once trained, the network can autonomously generate steering commands from video captured by a single center camera.

The network, consisting of 9 layers including 5 convolutional and 3 fully connected layers, is trained to minimize disparities between its steering commands and those of human drivers or adjusted commands for off-center and rotated images. Input images are split into YUV planes and normalized within the network. Strided convolutions in the first three layers and non-strided convolutions in the last two extract features. However, distinguishing between feature extraction and control is challenging due to end-to-end training. Data selection involves choosing frames labeled with road type, weather, and driver activity, sampled at 10 FPS to prevent redundancy.

Before on-road testing, the trained CNN's performance is evaluated in simulation. Utilizing pre-recorded videos and steering commands, the simulator adjusts images to reflect deviations from the ”ground truth” lane center. The CNN provides steering commands based on these adjusted images, guiding the virtual vehicle's position updates. Off-center distance, yaw, and distance traveled are recorded, with simulated human interventions triggered if the vehicle deviates over a set threshold.

Subsequently, the networks are assessed in two stages: simulation and on-road tests. In simulation, they provide steering commands for pre-recorded test routes covering diverse conditions in Monmouth County, NJ. The percentage autonomy, indicating the network's ability to drive without human intervention, is calculated based on simulated interventions. For instance, if there were 10 interventions in 600 seconds, the autonomy would be 90

VI. Our Model

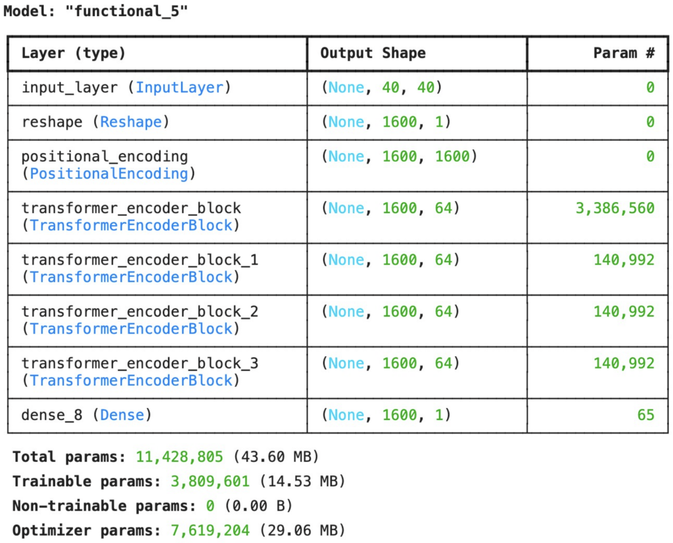
Our model aims to predict steering angles for autonomous vehicles using a novel approach that combines elements from the DAVE-2 system with ideas inspired by the Motion Transformer (MTR) and Multi-Granular Transformer (MGTR) methodologies. By leveraging the dataset from DAVE-2 and incorporating transformer-based architectures, we introduce a robust framework for accurate motion prediction.

A. Architecture

1. Input Layer: Represents the input images resized to a resolution of 40x40 pixels. These images serve as the primary input data for the model.

2. Reshape Layer: Reshapes the input images into a format suitable for further processing. The 1600 indicates the total number of pixels in the 40x40 images, with an additional dimension for channel depth.

3. Positional Encoding Layer: Adds positional information to the input data, allowing the model to capture spatial relationships between different pixels in the image. Positional encoding enhances the model's ability to understand the geometric structure of the input images.



Model: "functional_5"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 40, 40)	0
reshape (Reshape)	(None, 1600, 1)	0
positional_encoding (PositionalEncoding)	(None, 1600, 1600)	0
transformer_encoder_block (TransformerEncoderBlock)	(None, 1600, 64)	3,386,560
transformer_encoder_block_1 (TransformerEncoderBlock)	(None, 1600, 64)	140,992
transformer_encoder_block_2 (TransformerEncoderBlock)	(None, 1600, 64)	140,992
transformer_encoder_block_3 (TransformerEncoderBlock)	(None, 1600, 64)	140,992
dense_8 (Dense)	(None, 1600, 1)	65

```
Total params: 11,428,805 (43.60 MB)
Trainable params: 3,809,601 (14.53 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 7,619,204 (29.06 MB)
```

Fig. 3: Our model's Overview

4. Transformer Encoder Blocks: Consists of multiple Transformer Encoder blocks stacked on top of each other. Each Transformer Encoder block includes self-attention mechanisms, such as multi-head attention, followed by feed-forward neural networks (FFN). These blocks are responsible for learning representations of the input images, capturing important features for steering angle prediction.

5. Dense Layer: This dense layer serves as serves as the output layer, mapping learned features to the predicted steering angle, facilitating control of the vehicle's steering wheel.

B. Performance

Despite resource and memory constraints, our model demonstrates promising performance. On the test dataset, the model achieves an accuracy of 95.5%, we evaluated this by looking at the validation loss on the last epoch, which was 4.55%. While this accuracy may not match the state-of-the-art results achieved by Dave2 at 98% accuracy, it represents a significant advancement in motion prediction for autonomous vehicles.

VII. Limitations and Future Directions

While our model shows promise, there are areas for improvement. Future research could focus on optimizing the model architecture to improve accuracy further. Additionally, exploring techniques for mitigating resource and memory constraints could enhance the scalability and practicality of the model for real-world applications. Overall, our model lays a solid foundation for continued innovation in autonomous vehicle technology. We could further use the waymo dataset into this transformer model as deep learning models tend to give better accuracy with larger datasets.

VIII. Individual Contributions

Both researchers contributed to the project equally. Initially we explored various papers to understand the work done in this field in the past which is briefly mentioned in the related works section. Then we divided the main papers of MTR (Ali) and MGTR (Harsh) between ourselves. We looked at the whole architecture that went behind these frameworks. We then explored the Waymo dataset and set up the HPCs, but due to compute issues of Waymo being 1TB dataset we couldn't run the whole framework completely. Harsh then looked in the the Dave 2's simulation and ran the code on his HPC. We then researched on ways to implement the transformers in the Dave 2 data together and executed it successfully. We divided the report writing and the presentation between the stuff we worked more deeply towards. And overall had an equal distribution of tasks.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. (2016). Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. (2016). End to end learning for self-driving cars. *CoRR*, abs/1604.07316.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. *CoRR*, abs/2005.12872.

Casas, S., Luo, W., and Urtasun, R. (2021). Intentnet: Learning to predict intention from raw sensor data. *CoRR*, abs/2101.07907.

Gan, Y., Xiao, H., Zhao, Y., Zhang, E., Huang, Z., Ye, X., and Ge, L. (2024). Mgtr: Multi-granular transformer for motion prediction with lidar.

Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C., and Schmid, C. (2020). Vectornet: Encoding HD maps and agent dynamics from vectorized representation. *CoRR*, abs/2005.04259.

Marchetti, F., Becattini, F., Seidenari, L., and Bimbo, A. D. (2020). MANTRA: memory augmented networks for multiple trajectory prediction. *CoRR*, abs/2006.03340.

Park, S. H., Lee, G., Bhat, M., Seo, J., Kang, M., Francis, J., Jadhav, A. R., Liang, P. P., and Morency, L. (2020). Diverse and admissible trajectory forecasting through multimodal context understanding. *CoRR*, abs/2003.03212.

Shi, S., Jiang, L., Dai, D., and Schiele, B. (2023). Motion transformer with global intention localization and local movement refinement.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Zhao, H., Gao, J., Lan, T., Sun, C., Sapp, B., Varadarajan, B., Shen, Y., Shen, Y., Chai, Y., Schmid, C., Li, C., and Anguelov, D. (2021). Tnt: Target-driven trajectory prediction. In Kober, J., Ramos, F., and Tomlin, C., editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 895–904. PMLR.