2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Multi-Granular Transformer for Motion Prediction with LiDAR

Yiqian Gan*, Hao Xiao*, Yizhe Zhao*, Ethan Zhang, Zhe Huang, Xin Ye, Lingting Ge
TuSimple, Inc.

***Abstract*— Motion prediction has been an essential component of autonomous driving systems since it handles highly uncertain and complex scenarios involving moving agents of different types. In this paper, we propose a Multi-Granular TRansformer (MGTR) framework, an encoder-decoder network that exploits context features in different granularities for different kinds of traffic agents. To further enhance MGTR's capabilities, we leverage LiDAR point cloud data by incorporating LiDAR semantic features from an off-the-shelf LiDAR feature extractor. We evaluate MGTR on Waymo Open Dataset motion prediction benchmark and show that the proposed method achieved state-of-the-art performance, ranking 1st on its leaderboard [1].**

***Keywords*: Motion Prediction, Transformer, Autonomous Driving**

I. Introduction

High-quality motion prediction in a long horizon is essential for the development of safety-critical autonomous vehicles. It serves as one of the cornerstones of related fields including scene understanding and decision-making in the realm of autonomous driving. Although advancements were made in past years, major challenges still exist and come from the following aspects: (i) Heterogeneous data acquired by autonomous vehicles such as maps and agent history states is non-trivial to be represented in a unified space. (ii) Environment context inputs from upstream modules including object detection and pre-built maps have limitations (e.g., uncountable amorphous regions such as bushes, walls, and construction zones, can be missing). (iii) Multimodal nature of agent behaviors brings further complexity. Here, the multimodal agent behaviors refer to discrete and diverse agent intents and possible futures. This work addresses these challenges with a proposed multi-granular Transformer model, namely MGTR.

Early methods mainly render inputs including High Definition (HD) map, agent history states into rasterized images, and apply convolutional neural networks (CNN) to encode scene information [1]–[3]. While convenient, long-range interactions are hard to capture in rasterization-based methods due to the limited receptive field of convolutions. A majority of recent works represent inputs as vectors by pre-processing raw continuous inputs into discrete sample points at a fixed sample rate [4]–[6]. Such vectors are later sent to graph-structured models for scene-level information extraction. A low sample rate will lose geometric details like curb curvature. On the other end, a high sample rate requires models' stronger ability to learn complex topographic relationships like lane centerline connectivity, which constrains perception range due to an increased number of nodes with limited computational resources. As shown in Fig. 1, in reality, different types of agents with varying motion patterns would benefit from context information at multiple distinguished granularities. Concurrently, the computer vision community has developed multiple ways of multi-granularity processing, such as feature pyramid techniques [7]. Through the years, it has been proven to be effective in gaining more comprehensive context information.

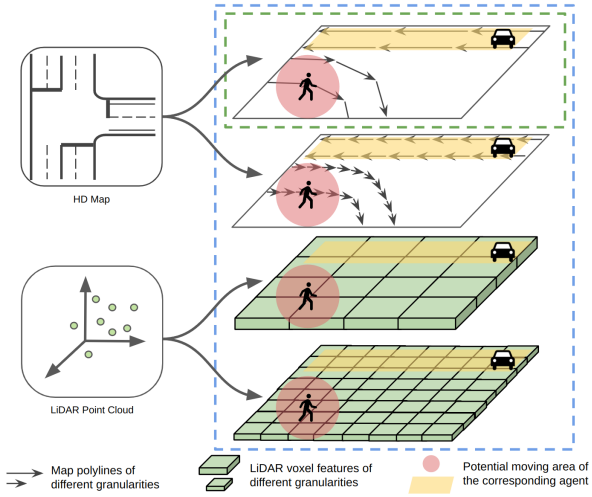

Fig. 1. **Comparing context information used in different motion prediction frameworks.** Most previous methods [5], [6] encode road graph only in a single granularity for all agents in the scene (**green** dashed box). In our method, various agents can benefit from multi-granular context information encoded from multimodal sources (**blue** dashed box).

Furthermore, the majority of existing motion prediction works are developed based on open datasets [8], [9] which usually provide 2D agent tracking states and static HD maps. However, in real world scenarios, a lot of other context information such as bushes, building walls, and traffic cones can also serve as strong cues for motion prediction. The new dataset WOMD-LiDAR [10] enables us to incorporate LiDAR point cloud containing dense 3D context information for motion prediction. To the best of our knowledge, only a few works have combined LiDAR data into motion prediction. They primarily focus on extracting instance-level information [10], [11] rather than rich environment context. Also, they do not explore LiDAR information in a multi-granular manner.

In this work, we propose a multi-granular Transformer model (MGTR), for motion prediction of heterogeneous traffic agents. MGTR follows a Transformer encoder-decoder architecture. It fuses multimodal inputs including agent his-

∗ equal contribution
[1]https://waymo.com/open/challenges/2023/motion-prediction/

tory states, map elements, and extra 3D context embeddings from LiDAR. Both map elements and LiDAR embeddings are processed into sets of tokens at several granular levels for better context learning. Next, agent embeddings and multi-granular context embeddings are passed through a Transformer encoder after being filtered by our motion-aware context search for better efficiency. Then, motion predictions are generated through iterative refinement within the decoder and modeled by Gaussian Mixture Model (GMM). Our contributions can be summarized as follows: (i) We introduce a novel Transformer-based motion prediction method utilizing multimodal and multi-granular inputs, with motion-aware context search mechanism to enhance accuracy and efficiency. (ii) We present an approach to incorporate LiDAR inputs practically and efficiently for the purpose of motion prediction. (iii) We demonstrate state-of-the-art performance on Waymo Open Dataset motion prediction benchmark.

II. Related work

Motion prediction: Early works [1]–[3], [12]–[17] on motion prediction usually represent inputs as rasterized images, and adopts CNN to obtain high-quality results. For more direct context representation, VectorNet [4] brings up the vector representation that sequentially samples and connects map and agent history states into polylines, and combines them with graph-based models. Vectorized inputs enable researchers to tackle both structured and unstructured data, and build more versatile models [5], [18], [19]. An emerging trend rises regarding Transformer-based models in various NLP and vision applications [20]. SceneTransformer [21] and Wayformer [22], combine the vector representation and apply a Transformer-based model to handle multimodal input. MTR [6] and MTR++ [23] further improve vectorized representation by using local connected graphs and apply Transformer structures with their inspirations from DETR [24] and DAB-DETR [25]. In this work, we adopt the vectorized representation but introduce the multi-granular structure for multimodal input, which is an essential aspect neglected by previous works.

Multi-granularity learning: The concept of multi-granularity learning originates from the field of computer vision. It is capable of learning patterns of various granular features [7], [26]–[29]. InceptionNet [26] and FPN [7] have shown great success in image classification tasks by incorporating multi-granularity techniques into CNN models, while MViT [29] proves the effectiveness of multi-granularity representation in Transformer-based models. In this work, we exploit advantages of applying multi-granularity to Transformer-based models for motion prediction, which has rarely been explored.

LiDAR for motion prediction: Applying LiDAR in the field of motion prediction is fairly new. In recent years, with the advancement of multimodal learning, researchers have been trying to incorporate LiDAR data towards different learning tasks in the field [10], [11], [30]–[32]. Most work such as IntentNet [30] and MultiXNet [31] take LiDAR data as the only perception input and generate both object detection and motion prediction through multi-task settings. More Recent works [10], [11] focus on motion prediction and mainly use LiDAR as instance-level features. Unlike previous work, the proposed MGTR treats LiDAR data as context features and incorporates the advantages of multi-modal learning.

III. Method

As depicted in Fig. 2, we proposed the MGTR model, a novel Transformer-based framework that takes multimodal inputs in a multi-granular manner including LiDAR data. In III-A, we first introduce how different inputs are represented and encoded into multi-granular tokens and how the number of tokens is reduced by motion-aware context search. Next, in III-B and III-C, we demonstrate how tokens are refined in the encoder and utilized in the decoder for motion prediction. Finally, III-D introduces the training losses used in our model.

A. Multimodal Multi-Granular Inputs

*1) **Agent and map**:* Following representation in VectorNet [4], agent state history is sampled at a constant time interval and processed into vectorized polylines as $\mathrm{P_A} \in \mathbb{R}^{N_a \times T_h \times C_a}$ to represent state information from $T_0 - T_h$ to T_0, where N_a denotes the number of target agents in a scene, C_a as the dimension of agent features, T_0 as the current time, T_h as the time horizon. The agent state features C_a include position, velocity, 3D bounding box size, heading angle, object type, etc. Zero paddings are added in the time dimension if the tracking length is smaller than T_h. Then, each agent polyline will first be transformed into the target agent-centric coordinate followed by a PointNet-like [33] polyline encoder as shown in Eq. 1.

Different types of agents have different movement ranges and requirements for map granularity. In this work we extract map contents in a multi-granular manner. Map elements with topological relationships such as road centerlines and area boundaries are sampled evenly at different sample rates, resulting in polylines with different granularities. Concretely, we represent sets of multi-granular polylines as $\{\mathrm{P}_\mathrm{M}^{(i)}\} \in \mathbb{R}^{N_m^{(i)} \times N_s^{(i)} \times C_m}$, where $\mathrm{P}_\mathrm{M}^{(i)}$ denotes map polylines at i-th spatial granularity, $N_m^{(i)}$ denotes the number of polylines at i-th granularity, $N_s^{(i)}$ denotes the number of sampled points in each polyline, C_m denotes the token feature dimension for map including positions, curvature, speed limit, etc. With a high sample rate, the same road centerline will be sampled into more polylines similar to more image pixels on high-resolution images. $N_m^{(i)}$ polylines are generated at each sample rate r_i. Similar to agent polylines, each map polyline is transformed to an agent-centric coordinate and encoded by a PointNet-like structure as:

$$F_\mathrm{A} = \phi\Big(\mathrm{MLP}\big(\Gamma(\mathrm{P_A})\big)\Big), \quad F_\mathrm{M}^{(i)} = \phi\Big(\mathrm{MLP}\big(\Gamma(\mathrm{P}_\mathrm{M}^{(i)})\big)\Big), \quad (1)$$

where $\Gamma(\cdot)$ denotes the coordinate transformation, $\mathrm{MLP}(\cdot)$ denotes a multi-layer perceptron, ϕ denotes max-pooling. Agent and map polylines are encoded into $F_\mathrm{A} \in \mathbb{R}^{N_a \times C}$

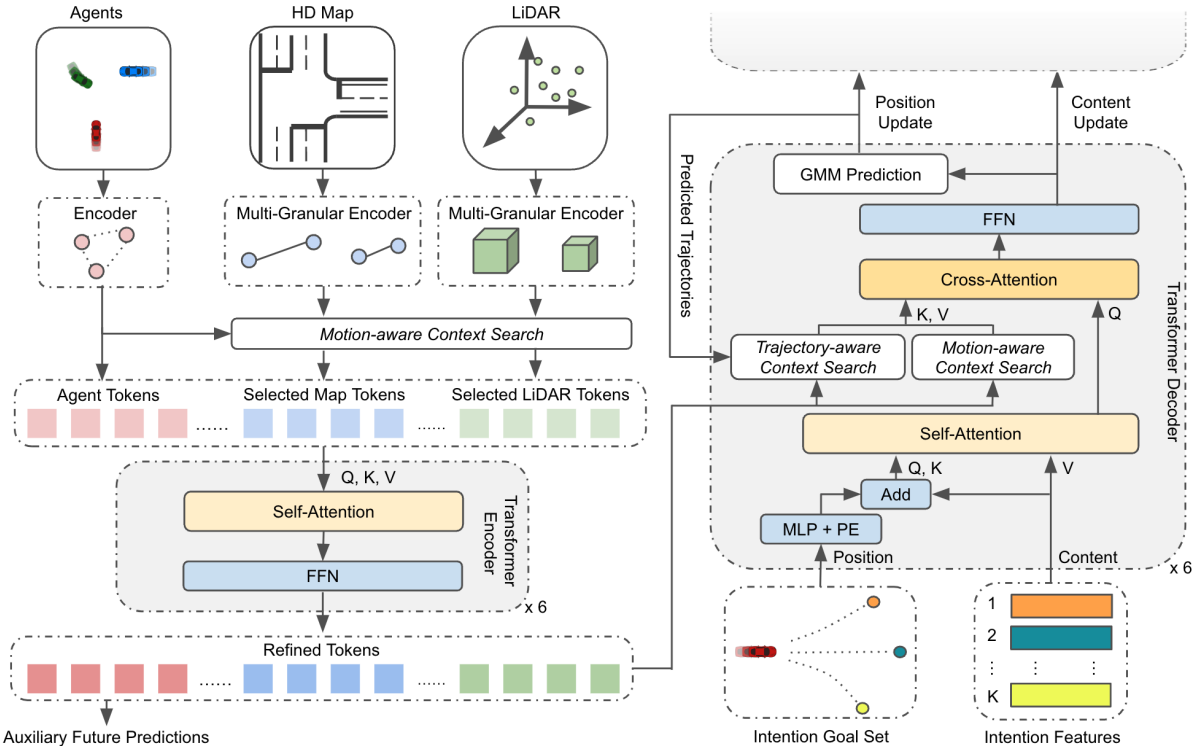

Fig. 2. **An overview of our proposed MGTR.** Agent trajectories and map elements are represented as polylines and encoded as agent and multi-granular map tokens. LiDAR data is processed by a pre-trained model into voxel features and further transformed into multi-granular LiDAR tokens. Motion-aware context search selects a set of map and LiDAR tokens, refined together with agent tokens through local self-attention in the Transformer encoder. Finally, a set of intention goals and their corresponding content features are sent to the decoder to aggregate context features. Multiple future trajectories of each agent will be predicted based on its intention goals, supporting the multimodal nature of agent behaviors.

and $F_{\mathrm{M}}^{(i)} \in \mathbb{R}^{N_m^{(i)} \times C}$ with a feature dimension of C. The weights of the polyline encoders for different granularities are not shared, to ensure map features at each granularity are kept.

*2) **LiDAR**:* In order to obtain richer 3D context information missing in explicit perception outputs and pre-built HD maps, we propose to integrate LiDAR information into our framework. Using raw LiDAR point cloud directly in the motion prediction network is inefficient and resource-intensive, due to its sparsity and large magnitude. Therefore, we choose to use LiDAR voxel features extracted by an off-the-shelf LiDAR segmentation network. This typically does not add additional overhead to autonomous driving systems since most deployed systems have such network already implemented. Although MGTR is not restricted to a certain LiDAR module, we adopt a voxel-based segmentation network, specifically LidarMultiNet [34], as our pre-trained LiDAR model. We extract a voxel feature map $\mathcal{V}_{raw} \in \mathbb{R}^{C_l \times D \times H \times W}$ from an intermediate layer, where C_l denotes the feature dimension, D, H, W are the sizes of the voxel space. It serves as a perfect input feature to represent context information for motion prediction. To add more context information, we concatenate one-hot embedding of the predicted semantic label of each voxel from the segmentation result and the 3D position of the center of each voxel with C_l LiDAR segmentation features. The voxel feature becomes $\mathcal{V} \in \mathbb{R}^{C_v \times D \times H \times W}$, where C_v is the LiDAR feature dimension after concatenation.

To obtain LiDAR context features in multi-granularities and reduce complexity, we employ average pooling across various scales to obtain features of different granularities. Pooled features are encoded into tokens through an MLP as:

$$F_{\mathrm{L}}^{(i)} = \mathrm{MLP}\Big(\mathcal{P}^{(i)}\big(\Gamma(\mathcal{V})\big)\Big), \tag{2}$$

where $\Gamma(\cdot)$ denotes the coordinate transformation, $\mathcal{P}^{(i)}(\cdot)$ denotes the average pooling for the i-th granularity. LiDAR features are encoded into $F_{\mathrm{L}}^{(i)} \in \mathbb{R}^{N_l^{(i)} \times C}$, where $N_l^{(i)}$ is the number of LiDAR tokens for the i-th granularity, and C is the feature dimension.

*3) **Motion-aware context search**:* After multi-granular encoders, the number of raw tokens $N = N_a + \sum_i N_m^{(i)} + \sum_i N_l^{(i)}$ can be extremely large, making it impossible to send them directly to Transformer encoder due to computing resource constraint. To learn features more efficiently, we introduce motion-aware context search which helps boost training efficiency and encode more meaningful context for agents with different motion patterns. For agents with different velocities, the desired positions of scene context for long-horizon trajectory prediction differ significantly. Therefore, for an agent of interest, we use its current velocity to project a future distance as the context token search prior. Through the projected position, we acquire $\widetilde{N}_m$ nearest map tokens and $\widetilde{N}_l$ nearest LiDAR tokens, resulting in a total of $N_a + \widetilde{N}_m + \widetilde{N}_l$ selected tokens that will be fed into our Transformer encoder for further refinement.

B. Transformer Encoder

*1) **Token aggregation and encoding**:* After the aforementioned vectorization and token generation, a Transformer encoder is established to aggregate features from multi-granular tokens. All tokens are refined through layers of encoder structure with a self-attention layer followed by a feed-forward network (FFN). To boost training efficiency and better capture neighboring information, a local attention mechanism is adopted. Let F_e^j be the refined tokens output by the j-th layer. The multi-head self-attention [35] can be formulated as:

$$\begin{aligned} Q &= F_e^{j-1} + PE(F_e^{j-1}), \quad V = \kappa(F_e^{j-1}), \\ K &= \kappa(F_e^{j-1}) + PE(\kappa(F_e^{j-1})), \\ F_e^j &= \text{MHSA}(Q, K, V), \end{aligned} \tag{3}$$

where $\kappa(\cdot)$ is a function that returns k-nearest neighboring tokens for each query token, $PE(\cdot)$ is a positional encoding function. $\text{MHSA}(\cdot,\cdot,\cdot)$ stands for multi-head self-attention layer [35].

*2) **Future state enhancement**:* In addition to considering agents' history trajectories, we also take into account their potential future trajectories, which play a crucial role in predicting the motion of agent of interest. Therefore, after agent tokens are refined by the Transformer encoder, a future trajectory is predicted for each agent following [6] and it can be formulated as:

$$\mathcal{T}_{scene} = \text{MLP}(F_e^A), \tag{4}$$

where $\mathcal{T}_{scene} \in \mathbb{R}^{N_a \times T \times 4}$ denotes trajectories (including position and velocity) of N_a agent for future T frames and F_e^A is the agent token from the encoder. The future trajectories are further encoded by a polyline encoder and fused with the original agent token F_e^A to form a future-aware agent feature that is fed into the decoder later. It's worth noting that $\mathcal{T}_{scene}$ is supervised by the ground truth trajectories, resulting in an auxiliary loss which will be introduced in III-D.

C. Transformer Decoder

*1) **Intention goal set**:* We generate $\mathcal{K}$ representative intention goals by adopting K-means clustering algorithm on endpoints of ground truth trajectories for different types of agents. Each intention goal represents an implicit motion mode, which can be modeled as a learnable positional embedding.

*2) **Token aggregation with intention goal set**:* In each layer, we first apply the self-attention module to propagate information among $\mathcal{K}$ intention queries as follows:

$$\begin{aligned} Q = K = F_d^{j-1} + PE(F_d^{j-1}), \quad V = F_d^{j-1}, \\ F_I^j = \text{MHSA}(Q, K, V), \end{aligned} \tag{5}$$

where $F_d^{j-1} \in \mathbb{R}^{\mathcal{K} \times C_{dec}}$ are intention features from $(j\text{-}1)$-th decoder layer and F_I^j is the updated intention feature, where $\mathcal{K}$ denotes number of intention goals, C_{dec} denotes feature dimension. We initialize the intention features F_d^0 to be all zeros as the input for the first Transformer decoder. Next, a cross-attention layer is adopted for aggregating features from the encoder as:

$$\begin{aligned} Q &= F_I^j + PE(F_I^j), \\ K = V &= \gamma(F_e) + PE(\gamma(F_e)), \\ F_d^j &= \text{MHCA}(Q, K, V), \end{aligned} \tag{6}$$

$$\gamma(F_e) = \eta(F_e) \cup \theta(F_e), \tag{7}$$

where F_I^j is the updated intention feature from previous multi-head self-attention. F_e is the multi-granular context tokens from encoder, which includes future-aware agent tokens, map tokens, and LiDAR tokens. $\text{MHCA}(\cdot,\cdot,\cdot)$ is the multi-head cross-attention layer [35]. $\gamma(\cdot)$ is the combination of our trajectory-aware context search ($\eta(\cdot)$) and motion-aware context search ($\theta(\cdot)$), which intends to extract multi-granular context features from a local region. Inspired by the dynamic map collection from [6], we introduce trajectory-aware context search which takes the predicted trajectories from previous decoder layer and selects the context token whose centers are close to the predicted trajectory. Along with the previously mentioned motion-aware context search, it continuously attends to the most important context information throughout iterative prediction refinement.

*3) **Multimodal motion prediction with GMM**:* Following [5], [6], we model the multimodal future trajectories with GMM. For each decoder layer, we append a classification head and a regression head for the intention feature F_d^j respectively:

$$p = \text{MLP}(F_d^j), \quad \mathcal{T}_{target} = \text{MLP}(F_d^j), \tag{8}$$

where $p \in \mathbb{R}^{\mathcal{K}}$ is the probability distribution of each trajectory mode corresponding to each intention goal. $\mathcal{T}_{target} \in \mathbb{R}^{\mathcal{K} \times T \times 7}$ is predicted GMM parameters representing $\mathcal{K}$ future trajectories and 2D velocities for T future frames. The endpoints of predicted trajectories will be used for positional embedding in the next decoder layer.

D. Training Loss

The training loss in this work is a weighted combination of: (i) auxiliary task loss $\mathcal{L}_{aux}$ on future predicted trajectories of all agents $\mathcal{T}_{scene}$ (ii) classification loss $\mathcal{L}_{cls}$ in form of cross entropy loss on predicted intention probability p (iii) GMM loss $\mathcal{L}_{GMM}$ in form of negative log-likelihood loss of the predicted trajectories of target agent $\mathcal{T}_{target}$. Auxiliary task loss is measured with L1 loss between ground truth and predictions of both agents' position and velocity. Similar to [6], we use a hard-assignment strategy that selects the best matching mode and calculates the GMM loss and the classification loss. This can force each mode to specialize for a distinct agent behavior.

IV. Experiments

A. Dataset

Following most recent motion prediction works [6], [10], [11], we conduct extensive experiments on WOMD-LiDAR dataset [10]. It is a large-scale motion dataset with LiDAR

specifically designed for motion prediction in the field of autonomous driving, containing 100,000+ real-world driving video clips with diverse driving scenarios (e.g. intersection, lane merging). It shares the same training, evaluation and test samples with its predecessor WOMD [36]. WOMD-LiDAR contains 3 categories of agents of interest: vehicle, pedestrian, and cyclist. The performance of this task on WOMD-LiDAR is measured by the minimum Average Displacement Error (minADE), the minimum Final Displacement Error (minFDE), miss rate (MR) and the mean Average Precision (mAP) of predicted trajectories in the next 3, 5 and 8 seconds, with mAP as the major evaluation metric. Details of those metric definitions can be found in [36].

B. Experiment Setup

*1) **LiDAR voxel encoder**:* We use LidarMultiNet [34] as our LiDAR encoder to extract LiDAR voxel features from raw point cloud. Resulting 3D voxel features after its global context pooling are used as our prediction model input. The original feature dimension C_l is 32. As discussed in Sec III-A.2, for each voxel, we concatenate a 22-dim predicted one-hot segmentation result as well as its 3-dim position by channel, formulating our 57-dim LiDAR voxel features (i.e. $C_v = 57$). Average pooling is applied twice on original voxels to form multi-granularities, making the length and width of each LiDAR voxel 0.8m or 1.6m. LidarMultiNet is pre-trained on Waymo Open Dataset [37] and is frozen during the training of our model.

*2) **Implementation details**:* For full-scale experiments on WOMD-LiDAR `val` set and `test` set, our batch size is 10 per GPU and we train from scratch for 30 epochs. We use AdamW optimizer [38] with an initial learning rate of 0.0001, in conjunction with a multi-step scheduler. The learning rate is decayed by a factor of 0.5 every two epochs after 20 epochs. Both the encoder and the decoder consist of 6 Transformer layers. We adopt $\widetilde{N}_m = 768$ map polylines as the topographic context in the encoder. In addition, we add $\widetilde{N}_l = 256$ multi-granular LiDAR voxels to complement our local scene representation. In practice, the length of each map polyline is either 10 map points or 20 map points, equivalent to 5m or 10m in range. The number of neighbors in local self-attention of the encoder is set to 32. For each sample, we predict 64 candidate trajectories using 64 intention goals, which are generated from K-means clustering over all ground truth goal points (at 8-second prediction horizon) in the training set. Non-maximum suppression (NMS) is applied to post-process predictions, resulting in 6 final trajectories per sample.

For ablation study, unless stated otherwise, all experiments use a same hyperparameter set as the full-scale experiments. We only use 20% of the total training data in WOMD-LiDAR via a fixed sampler for efficiency purposes.

C. Quantitative Analysis

We report quantitative results on WOMD-LiDAR `val` set, as shown in TABLE I. We compare MGTR against other state-of-the-art models. Remarkably, we achieve the best mAP over all types of agents. Specifically, compared to Wayformer+LiDAR [10], a multimodal model with LiDAR input, our model substantially improves mAP on pedestrians by 7%, cyclists by 4%, and vehicle by 1% for $t = 8$ seconds. MGTR demonstrates similar advancement compared with previous SOTA MTR [6] and MTR++ [23]. We argue that this indicates MGTR is better at capturing subtle movements thanks to the multi-granular representation of both map and LiDAR.

TABLE I

COMPARISON ON WOMD-LIDAR VAL SET. RESULTS IN TOP THREE ROWS ARE COMPUTED AS AN AVERAGE OF $t = 3, 5$, AND 8 SECONDS, WHILE THE ONES IN BOTTOM THREE ROWS ARE REPORTED FOR $t = 8$ SECONDS. MTR++ [23] DOES NOT REPORT CATEGORICAL RESULTS. FOLLOWING [10], ALL METRICS ARE REPORTED WITH TWO DECIMAL PLACES. $*$ INDICATES METHODS UTILIZING LIDAR.

	mAP ↑			
Method	Vehicle	Pedestrian	Cyclist	Average
MTR [6]	0.45	0.44	0.36	0.42
MTR++ [23]	-	-	-	0.44
MGTR* (Ours)	**0.46**	**0.47**	**0.40**	**0.45**
Wayformer [22]	0.35	0.35	0.29	0.33
Wayformer+LiDAR [10]*	0.37	0.37	0.28	0.34
MGTR* (Ours)	**0.38**	**0.44**	**0.32**	**0.38**

Furthermore, we achieve the state-of-the-art performance on WOMD-LiDAR `test` set. As revealed in TABLE II, the single-model version of our proposed MGTR achieves an overall mAP of 45.05%, which has +1.76% advantage over the second-best model, significantly outperforming all other single-model entries. Compared to the latest state-of-the-art motion prediction model, MTR++ [23], we achieve a whopping 5.41% increase in terms of mAP on the pedestrian category. Apart from mAP, our model also improves over a variety of metrics, including minADE, minFDE, MR for multiple categories. Specifically, we reduce trajectory miss rate (MR) on cyclists by 1.11% in comparison with the second-best model. This strongly signals that for non-vehicular objects, features that attend to details are key to more accurate and reliable trajectory predictions.

As of the paper submission date (Sep. 15, 2023), it is worth noting that our proposed MGTR with model ensemble has ranked **first place** among all submissions on the motion prediction track of Waymo Open Challenge Leaderboard [2]. Detailed comparison of methods with model ensemble techniques is not included in the paper due to page limits.

D. Qualitative Analysis

We further delve into the qualitative aspects of our proposed MGTR model using visualizations. While our model is capable of generating multimodal trajectories, for the sake of illustration, we have chosen to visualize the trajectory with the highest probability, demonstrating the efficacy of

[2]https://waymo.com/open/challenges/2023/motion-prediction/

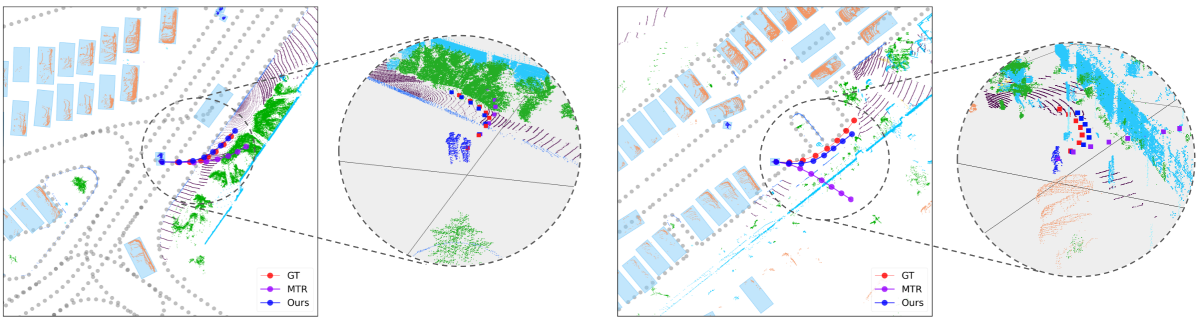

Fig. 3. **Visualization of prediction result comparison between MTR [23] and MGTR (Ours)**. A global bird's-eye-view (including agents, HD map and LiDAR point cloud) and a local LiDAR visualization for each scene. For LiDAR point cloud, only limited semantic class such as vegetation (**green** points), building (**cyan** points), sidewalk (**brown** points), vehicle(**orange** points) and pedestrian (**blue** points) are shown for better visualization.

TABLE II

COMPARISON ON WOMD-LIDAR TEST SET. ALL METRICS ARE AVERAGED OVER 3S, 5S, AND 8S. ALL MODELS DO NOT USE MODEL ENSEMBLE.

	Vehicle				Pedestrian				Cyclist				Avg
Method	**minADE↓**	**minFDE↓**	**MR↓**	**mAP↑**	**minADE↓**	**minFDE↓**	**MR↓**	**mAP↑**	**minADE↓**	**minFDE↓**	**MR↓**	**mAP↑**	**mAP↑**
ReCoAt [39]	0.9865	2.1771	0.2695	0.2667	0.4261	0.8982	0.1451	0.3208	0.8985	1.9252	0.3164	0.2258	0.2711
DenseTNT [19]	1.3462	1.9120	0.1518	0.3698	0.5013	0.9130	0.1014	0.3342	1.2687	1.8292	0.2186	0.2802	0.3281
SceneTransformer [21]	**0.7094**	**1.4115**	0.1480	0.3270	0.3812	0.7532	0.0971	0.2715	0.7446	1.4701	0.2239	0.2380	0.2788
GTR-R36 [40]	0.7450	1.5049	0.1477	0.4521	0.3470	0.7221	0.0741	0.4243	0.7095	1.4406	0.1772	0.4003	0.4255
DM [41]	0.7701	1.5400	0.1529	0.4725	0.3741	0.7882	0.0848	0.4172	0.7436	1.4885	0.2043	0.4005	0.4301
MTR [6]	0.7642	1.5257	0.1514	0.4494	0.3486	0.7270	0.0753	0.4331	0.7022	**1.4093**	0.1786	0.3561	0.4129
MTR++ [23]	0.7178	1.4321	**0.1366**	**0.4871**	0.3504	0.7305	0.0745	0.4324	0.7036	1.4190	0.1784	0.3792	0.4329
MGTR (Ours)	0.7393	1.5119	0.1497	0.4626	**0.3441**	**0.7191**	**0.0722**	**0.4865**	**0.6919**	1.4096	**0.1675**	**0.4023**	**0.4505**

our approach. Fig. 3 showcases two scenarios in which a pedestrian is crossing a street. When we examine the trajectory predicted by MTR [23], in the scenarios the pedestrian will (a) walk into the bushes and (b) pass through a building, which are not reasonable outcomes. In contrast, our MGTR model successfully predicts that the pedestrian will walk onto the sidewalk, skillfully (a) avoiding any collision with the bushes, and (b) avoiding the building. These visualizations underscore the superior predictive capabilities of MGTR in comparison to the MTR, particularly in situations where LiDAR can provide additional 3D context information that is not available within HD map.

E. Ablation Study

We conduct ablation studies on WOMD-LiDAR *val* set, as shown in TABLE III. The baseline has a similar structure as MGTR, but only possesses single coarse-granular map tokens. By adding our proposed designs one by one, we observe consistent and notable mAP improvements for all types of agents.

*1) **Multi-granular map**:* The multi-granular design allows our model to represent the world at different resolutions so that agents can benefit from the granularity that suits them. Compared to the baseline, adding multi-granularity significantly improves mAP of all types, especially for pedestrians.

*2) **Multi-granular LiDAR**:* Voxelized LiDAR features contain fine-grained scene information that can affect agents' future behaviors but may not be covered by perception results and HD map. Adopting LiDAR as context features improves mAP of pedestrians and cyclists drastically.

TABLE III

ABLATION STUDY ON OUR PROPOSED MGTR.

	mAP ↑			
Description	Vehicle	Pedestrian	Cyclist	Average
Baseline	0.3860	0.3682	0.2881	0.3474
+ multi-granular map	0.3895	0.3730	0.2900	0.3508
+ multi-granular LiDAR	0.3896	0.3820	0.2997	0.3571
+ motion-aware context search	**0.3919**	**0.3935**	**0.3025**	**0.3626**

*3) **Motion-aware context search**:* Compared to the original context search introduced in [23] that does not consider individual moving patterns, our motion-aware context search factors in motion compensation on an individual basis. After adding this feature, we observe significant improvements on mAP. It indicates that our design can better accommodate different agents of interest to retrieve relevant context information, thus improving motion prediction accuracy.

V. CONCLUSION

In this paper, we propose MGTR, a novel Transformer-based motion prediction model that incorporates multimodal inputs including LiDAR point cloud in an effective multi-granular manner. Rich context features at different granularities enhance the overall motion prediction performance. Our model reaches state-of-the-art performance on the public WOMD-LiDAR motion prediction benchmark with significant improvements over pedestrians and cyclists.

REFERENCES

[1] L. Fang, Q. Jiang, J. Shi, and B. Zhou, "Tpnet: Trajectory proposal network for motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6797–6806.

[2] P. Wu, S. Chen, and D. N. Metaxas, "Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 385–11 395.

[3] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2090–2096.

[4] 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," 2020.

[5] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov *et al.*, "Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7814–7821.

[6] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion transformer with global intention localization and local movement refinement," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6531–6543, 2022.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[8] 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," 2021.

[9] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[10] 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," 2023.

[11] 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," 2023.

[12] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, and J. Schneider, "Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2095–2104.

[13] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.

[14] F. Marchetti, F. Becattini, L. Seidenari, and A. D. Bimbo, "Mantra: Memory augmented networks for multiple trajectory prediction," 2021.

[15] S. H. Park, G. Lee, M. Bhat, J. Seo, M. Kang, J. Francis, A. R. Jadhav, P. P. Liang, and L.-P. Morency, "Diverse and admissible trajectory forecasting through multimodal context understanding," 2020.

[16] S. Casas, C. Gulino, R. Liao, and R. Urtasun, "Spatially-aware graph neural networks for relational behavior forecasting from sensor data," 2019.

[17] S. Casas, A. Sadat, and R. Urtasun, "Mp3: A unified model to map, perceive, predict and plan," 2021.

[18] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, C. Li, and D. Anguelov, "Tnt: Target-driven trajectory prediction," 2020.

[19] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," 2021.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[21] 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.

[22] 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.

[23] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying," *arXiv preprint arXiv:2306.17770*, 2023.

[24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020.

[25] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "Dab-detr: Dynamic anchor boxes are better queries for detr," *arXiv preprint arXiv:2201.12329*, 2022.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[27] H. Xiao, W. Lin, B. Sheng, K. Lu, J. Yan, J. Wang, E. Ding, Y. Zhang, and H. Xiong, "Group re-identification: Leveraging and integrating multi-grain information," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 192–200.

[28] W. Lin, Y. Li, H. Xiao, J. See, J. Zou, H. Xiong, J. Wang, and T. Mei, "Group reidentification with multigrained matching and integration," *IEEE transactions on cybernetics*, vol. 51, no. 3, pp. 1478–1492, 2019.

[29] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6824–6835.

[30] S. Casas, W. Luo, and R. Urtasun, "Intentnet: Learning to predict intention from raw sensor data," in *Conference on Robot Learning*. PMLR, 2018, pp. 947–956.

[31] N. Djuric, H. Cui, Z. Su, S. Wu, H. Wang, F.-C. Chou, L. San Martin, S. Feng, R. Hu, Y. Xu *et al.*, "Multixnet: Multiclass multistage multimodal motion prediction," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 435–442.

[32] A. Laddha, S. Gautam, S. Palombo, S. Pandey, and C. Vallespi-Gonzalez, "Mvfusenet: Improving end-to-end object detection and motion forecasting through multi-view fusion of lidar data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2865–2874.

[33] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2017.

[34] 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," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 3231–3240.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[36] 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.

[37] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[38] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[39] Z. Huang, X. Mo, and C. Lv, "Recoat: A deep learning-based framework for multi-modal motion prediction in autonomous driving application," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 988–993.

[40] L. Haochen, M. Xiaoyu, H. Zhiyu, and L. Chen, "Transformer with group-wise modal assignments for motion prediction," 2023.

[41] Y. Ting, J. Lingxin, and L. Wei, "Dmp: Destination-driven motion prediction with prior fusion," 2023.