# GANet: Goal Area Network for Motion Forecasting Technical Report

**Mingkun Wang**
Peking University
wangmingkun95@qq.com

**Changqian Yu**
Meituan
yuchangqian@meituan.com

**Mingxu Wang**
Fudan University
wangmingxu99@qq.com

**Dongchun Ren**
Meituan
rendongchun@meituan.com

**Deheng Qian**
Meituan
qiandeheng@meituan.com

## Abstract

Predicting the future motion of road participants is crucial for autonomous driving but is extremely challenging due to staggering motion uncertainty. Recently, most motion forecasting methods resort to the goal-based strategy, i.e., predicting endpoints of motion trajectories as conditions to regress the entire trajectories, so that the search space of solution can be reduced. However, accurate goal coordinates are hard to predict and evaluate. In addition, the point representation of the destination limits the utilization of a rich road context, leading to inaccurate prediction results in many cases. Goal area, i.e., the possible destination area, rather than goal coordinate, could provide a more soft constrain for searching potential trajectories by involving more tolerance and guidance. In view of this, we propose a new goal area-based framework, named Goal Area Network (GANet), for motion forecasting, which models goal areas rather than exact goal coordinates as preconditions for trajectory prediction, performing more robustly and accurately. Specifically, we propose a GoICrop (Goal Area of Interest) operator to effectively extract semantic lane features in goal areas and model actors' future interactions, which benefits a lot for future trajectory estimations.

## 1 Introduction

We propose Goal Area Network framework (GANet) that predicts potential goal areas as conditions for motion forecasting. As shown in Figure 1, there are three stages in GANet, which are trained in an end-to-end way, and we construct a series of GANet models following this framework. They overcome the shortcomings of the aforementioned goal-based prediction methods. First, an efficient encoding backbone is adopted to encode motion history and scene context. Then, we predict approximate goals and crop their surrounding goal areas as more robust conditions. Moreover, we introduce a GoICrop operator to explicitly query and aggregate the rich semantic features of lanes in the goal areas instead of using stuffless goal coordinates embeddings. GoICrop learns the interactions between maps and actors in the goal areas, and captures the interactions among actors in the future implicitly. Finally, we make the formal motion forecasting conditioned on motion history, scene context, and the aggregated goal area features.

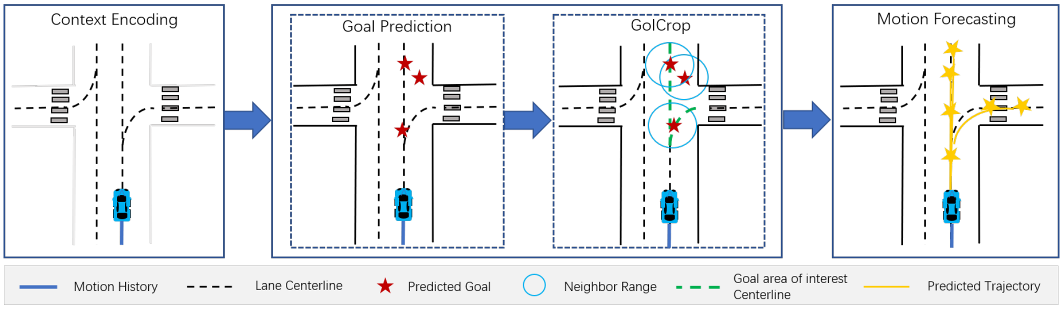

Figure 1: Illustration of GANet framework, which consists of three stages: (a) Context encoding encodes motion history and scene context; (b) Goal prediction predicts possible goals. GoICrop retrieves and aggregates goal area map features and models the actors' interactions in the future; (c) Motion forecasting estimates multi-feasible trajectories and their corresponding confidence scores.

Our method is different from previous works as follows. (1) We make the first attempt to propose a new goal area-based framework with three stages trained end-to-end for motion forecasting, which makes predictions based on motion history, scene context, and predicted goal areas. We give the definition of goal areas and experimentally verify the effectiveness of modeling goal areas. (2) We employ a GoIcrop operator to extract rich semantic map features in goal areas, which effectively models distant relevant map features slighted by previous methods. These map features provide more robust information than the goal coordinates embedding. This is because that GoICrop's distance-based attention implicitly captures the interactions between maps and trajectories in goal areas and the interactions among actors in the future. This also constrains the trajectories to follow driving rules and map topology in a data-driven manner, rather than relying on a well-designed goal space.

## 2 Method

This section describes our proposed GANet framework in a pipelined manner. An overview of the GANet model architecture is shown in Fig. 2.

### 2.1 Motion history and scene context encoding

As shown in Figure 2, the first stage of motion forecasting is driving context encoding, which extracts actors' motion features and maps features, and models their interactions. We adopt LaneGCN's [4] backbone to encode motion history and scene context for its outstanding performance. Specifically, we apply a 1D CNN with Feature Pyramid Network (FPN) to extract actors' motion features. The input is observed past trajectories of all actors in a scenario. We represent each trajectory as a sequence of Bird's Eye View (BEV) coordinate displacements $\{\Delta p_{-(T'+1)}, ..., \Delta p_{-1}, \Delta p_0\}$, where $\Delta p_t$ is the 2D coordinates displacements from step $t-1$ to $t$. We observe $T'$ steps. For trajectories with observed steps less than $T'$, we pad them with zeros by adding a binary $1 \times T'$ mask to indicate whether the element is padded or not. We concatenate the displacements and the mask to obtain an input tensor of size $3 \times T'$.

Following [4], we use a multi-scale LaneConv network to encode map features. After driving context encoding, we obtain a 2D feature matrix $X$ where each row $X_i$ indicates the feature of the $i$-th actor, and a 2D feature matrix $Y$ where each row $Y_i$ indicates the feature of the $i$-th lane node. We can also use other methods to encode motion history and scene context.

### 2.2 Goal prediction

In stage two, we predict possible goals for the $i$-th actor based on $X_i$. In practice, a driver's driving intent is highly multi-modal. For example, he or she may stop, go ahead, turn left, or turn right when approaching an intersection. Therefore, we try to make a multiple-goals prediction. We construct a goal prediction header with two branches to predict $M$ possible goals and their confidence scores for each actor. We apply a Multi Layer-Perception Neural Network (MLP) in the regression branch to

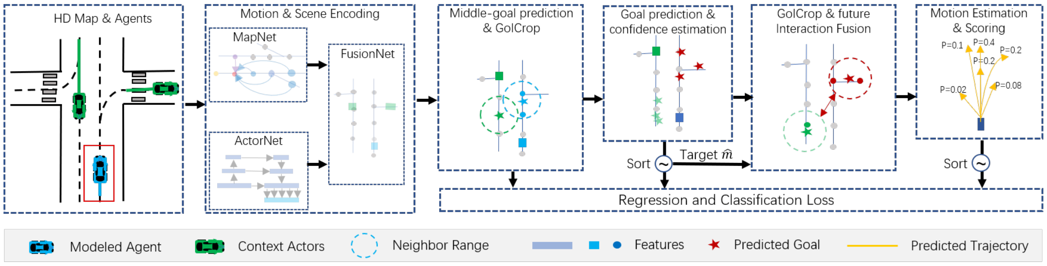

Figure 2: The GANet model overview. (a) A feature extracting model encodes and fuses map and motion features. (b) The "one goal prediction" module predicts a goal area in the trajectory's middle position and aggregates its features. (c) The "three goals predictions" module predicts three goal areas, aggregates their features, and models the actors' future interactions. (d) The final prediction stage predicts $K$ trajectories and their confidence scores.

regress $M$ BEV coordinates $G_{n,end} = \{(g^m_{n,end})\}_{j\in[0,M-1]}$, where $g^m_{n,end}$ is the $m$-th predicted goal coordinates of the $n$-th actor. For the classification branch, we apply an MLP to output $M$ confidence scores $C_{n,end} = \{(c^m_{n,end})\}_{m\in[0,M-1]}$, where $c^m_{n,end}$ is the $m$-th predicted goal confidence of the $n$-th actor. We use the sum of classification loss and regression loss to train this stage.

Given $M$ predicted goals, we find a positive goal $\hat{m}$ that has the minimum Euclidean distance with the ground truth coordinates at the final step. For classification, we use the max-margin loss similar to [4]:

$$L_{cls\_end} = \frac{1}{N(M-1)} \sum_{n=1}^{N} \sum_{m\neq\hat{m}} max(0, c^m_{n,end} + \epsilon - c^{\hat{m}}_{n,end}) \tag{1}$$

where $N$ is the total number of actors and $\epsilon = 0.2$ is the margin. The margin loss expects each goal to capture a specific pattern and pushes the goal closest to the ground truth to have a highest score. For regression, we apply the smooth L1 loss on the positive goals:

$$L_{reg\_end} = \frac{1}{N} \sum_{n=1}^{N} reg(g^{\hat{m}}_{n,end} - a^*_{n,end}) \tag{2}$$

where $a^*_{n,end}$ is the ground truth BEV coordinates of the $n$-th actor trajectory's final step, $reg(z) = \sum_i d(z_i)$, $z_i$ is the $i$-th element of $z$, and $d(z_i)$ is a smooth L1 loss.

Additionally, we also try to add a "one goal prediction" module at each trajectory's middle position to retrieve the map features at the approximate middle position. The loss term for this module is given by:

$$L_{reg\_mid} = \frac{1}{N} \sum_{n=1}^{N} reg(g_{n,mid} - a^*_{n,mid}) \tag{3}$$

where $a^*_{n,mid}$ is the ground truth BEV coordinates of the $n$-th actor trajectory's middle step.

The total loss at the goal prediction stage is:

$$L_1 = \alpha_1 L_{cls\_end} + \beta_1 L_{reg\_end} + \rho_1 L_{reg\_mid} \tag{4}$$

### 2.3 GoICrop

We choose the predicted goal with the highest confidence among $M$ goals as an anchor. This anchor is the approximate destination with the highest possibility that the actor may reach based on its motion history and driving context. We crop maps within 6 meters of the anchor as the goal area of interest rather than accurate goal coordinates, which relaxes the strict goal prediction requirement. The future behavior of an actor strongly depends on its destination area's context, i.e., the maps and other actors. Although previous works have explored the interactions between actors, the interactions between actors and maps in goal areas and the interactions among actors in the future have received less attention. Thus, we retrieve lane nodes in goal areas and apply a GoICrop module to aggregate these map node features as follows:

$$x_i' = \phi_1(x_i W_0 + \sum_j \phi_2(concat(x_i W_1, \Delta_{i,j}, y_j)W_2))W_3 \tag{5}$$

where $x_i$ is the feature of $i$-th actor and and $y_j$ is the feature of $j$-th lane node, $W_i$ is a weight matrix, $\phi_i$ is a layer normalization with ReLU function, and $\Delta_{i,j} = \phi(MLP(v_i - v_j))$, where $v_i$ denotes the anchor's coordinates of $i$-th actor and $v_j$ denotes the $j$-th lane node's coordinates. GoICrop serves as spatial distance-based attention and updates the goal area lane nodes' features back to the actors. We transpose $x_i$ with $W_1$ as a query embedding. The relative distance feature between the anchor of $i$-th actor and $j$-th lane node are extracted by $\Delta_{i,j}$. Then, we concatenate the query embedding, relative distance feature, and the lane node feature. An $MLP$ is employed to transpose and encode these features. Finally, the goal area features are aggregated for $i$-th actor.

Previous motion forecasting methods usually focus on the interactions in the observation history. However, actors will interact with each other in the future to follow driving etiquette, such as avoiding collisions. Since we have performed predictive goal predictions and gotten possible goals for each actor, our framework can model the actors' future interactions. Hence, we utilize the predicted anchor positions and apply a GoICrop module as equation 5 to implicitly model actors' interactions in the future. We consider the other actors whose future anchor's $l_2$ distance from the anchor of $i$-th actor are samller than 100 meters.

### 2.4 Motion estimation and scoring

We take the updated actor features $X$ as input to predict $K$ final future trajectories and their confidence scores in stage three. Specifically, we construct a two-branch multi-modal prediction header similar to the goal prediction stage, with one regression branch estimating the trajectories and one classification branch scoring the trajectories. For each actor, we regress K sequences of BEV coordinates $A_{n,F} = \{(a_{n,1}^k, a_{n,2}^k, ..., a_{n,T}^k)\}_{k\in[0,K-1]}$, where $a_{n,t}^k$ denotes the $n$-th actor's future coordinates of the $k$-th mode at $t$-th step. For the classification branch, we output $K$ confidence scores $C_{n,cls} = \{(c_n^k)\}_{k\in[0,K-1]}$ corresponding to $K$ modes. We find a positive trajectory of mode $\hat{k}$, whose final-step coordinate has the minimum Euclidean distance with the ground truth. For classification loss $L_{cls}$, we use the margin loss similar to the goal prediction stage. For regression loss $L_{reg}$, we apply the smooth L1 loss on all predicted steps of the positive trajectories.

To emphasize the importance of the goal, we add a loss term $L_{end}$ stressing the penalty at the final step.

The loss function for training at this stage is given by:

$$L_2 = \alpha_2 L_{cls} + \beta_2 L_{reg} + \rho_2 L_{end} \tag{6}$$

### 2.5 Training and inference

As all the modules are differentiable, we train our model with the loss function:

$$L = L_1 + L_2 \tag{7}$$

Although the various losses may seem complicated, their structures are almost the same. The parameters are chosen to balance the training process.

For inference, GANet (1) encodes motion and scene context; (2) predicts a goal at the middle position, crops a goal area, and aggregates its map features; (3) predicts three goals and chooses the one with the highest confidence to crop a goal area of interest; (4) aggregates the goal area map features and models actors' interactions in the future; (5) estimates $K$ trajectories and their confidence scores.

## 3 Implementation Details

We train our model on 2 A100 GPUs using a batch size of 128 with the Adam optimizer for 42 epochs. The initial learning rate is 1 x 10-3, decaying to 1 x 10-4 at 32 epochs.

Table 1: Results on Argoverse1 motion forecasting test (leaderboard).

| Method | b-minFDE (K=6) | MR (K=6) | minFDE (K=6) | minADE (K=6) | minFDE (K=1) | minADE (K=1) | MR (K=1) |
|---|---|---|---|---|---|---|---|
| LaneRCNN [1] | 2.147 | 0.123 | 1.453 | 0.904 | 3.692 | 1.685 | 0.569 |
| TNT[2] | 2.140 | 0.166 | 1.446 | 0.910 | 4.959 | 2.174 | 0.710 |
| DenseTNT (MR)[3] | 2.076 | 0.103 | 1.381 | 0.911 | 3.696 | 1.703 | 0.599 |
| LaneGCN [4] | 2.059 | 0.163 | 1.364 | 0.868 | 3.779 | 1.706 | 0.591 |
| mmTransformer[6] | 2.033 | 0.154 | 1.338 | 0.844 | 4.003 | 1.774 | 0.618 |
| GOHOME [8] | 1.983 | 0.105 | 1.450 | 0.943 | 3.647 | 1.689 | 0.572 |
| HOME [7] | - | **0.102** | 1.45 | 0.94 | 3.73 | 1.73 | 0.584 |
| DenseTNT (FDE)[3] | 1.976 | 0.126 | 1.282 | 0.882 | 3.632 | 1.679 | 0.584 |
| TPCN [5] | 1.929 | 0.133 | 1.244 | 0.815 | 3.487 | **1.575** | 0.560 |
| **GANet(Ours)** | **1.790** | 0.118 | **1.161** | **0.806** | **3.455** | 1.592 | **0.550** |

Table 2: Results on Argoverse2 motion forecasting test (leaderboard).

| Method | b-minFDE (K=6) | MR (K=6) | minFDE (K=6) | minADE (K=6) | minFDE (K=1) | minADE (K=1) | MR (K=1) |
|---|---|---|---|---|---|---|---|
| GANet | 1.9791 | **0.1709** | **1.3614** | 0.7335 | **4.5752** | 1.8078 | 0.6082 |

# 4 Experiments

## 4.1 Comparison with State-of-the-art

We compare our approach with state-of-the-art methods. As shown in Table 1, our GANet outperforms existing goal-based approaches. Specifically, we make a detailed comparison with LaneGCN because we adopt their backbone to encode motion history and scene context, which demonstrate the effectiveness of GANet. Public results on the official motion forecasting challenge leaderboard show that our GANet method significantly beats LaneGCN by decreases of 28%, 15%, 13% and 9% in MR6, minFDE6, brier-minFDE6, and minFDE1, respectively. Table 2 shows our result on argoverse2 test set.

## References

[1] Zeng, W., Liang, M., Liao, R., & Urtasun, R. (2021). Lanercnn: Distributed representations for graph-centric motion forecasting. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 532-539). IEEE.

[2] Zhao, H., Gao, J., Lan, T., Sun, C., Sapp, B., Varadarajan, B., ... & Anguelov, D. (2020). Tnt: Target-driven trajectory prediction. arXiv preprint arXiv:2008.08294.

[3] Gu, J., Sun, C., & Zhao, H. (2021). Densetnt: End-to-end trajectory prediction from dense goal sets. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 15303-15312).

[4] Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., & Urtasun, R. (2020, August). Learning lane graph representations for motion forecasting. In European Conference on Computer Vision (pp. 541-556). Springer, Cham.

[5] Ye, M., Cao, T., & Chen, Q. (2021). Tpcn: Temporal point cloud networks for motion forecasting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11318-11327).

[6] Liu, Y., Zhang, J., Fang, L., Jiang, Q., & Zhou, B. (2021). Multimodal motion prediction with stacked transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7577-7586).

[7] Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., & Moutarde, F. (2021, September). Home: Heatmap output for future motion estimation. In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC) (pp. 500-507). IEEE.

[8] Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., & Moutarde, F. (2021). Gohome: Graph-oriented heatmap output for future motion estimation. arXiv preprint arXiv:2109.01827.

[9] Huang, Z., Mo, X., & Lv, C. (2021). Multi-modal Motion Prediction with Transformer-based Neural Network for Autonomous Driving. arXiv preprint arXiv:2109.06446.

[10] Chang, M. F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., ... & Hays, J. (2019). Argoverse: 3d tracking and forecasting with rich maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8748-8757).

[11] Helbing, D., & Molnar, P. (1995). Social force model for pedestrian dynamics. Physical review E, 51(5), 4282.

[12] Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016). Social lstm: Human trajectory prediction in crowded spaces. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 961-971).

[13] Zhang, P., Ouyang, W., Zhang, P., Xue, J., & Zheng, N. (2019). Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 12085-12094).

[14] Wang, M., Shi, D., Guan, N., Zhang, T., Wang, L., & Li, R. (2019, November). Unsupervised pedestrian trajectory prediction with graph neural networks. In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 832-839). IEEE.

[15] Mercat, J., Gilles, T., El Zoghby, N., Sandou, G., Beauvois, D., & Gil, G. P. (2020, May). Multi-head attention for multi-modal joint vehicle motion forecasting. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 9638-9644). IEEE.

[16] Chai, Y., Sapp, B., Bansal, M., & Anguelov, D. (2019). Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. arXiv preprint arXiv:1910.05449.

[17] Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C., & Schmid, C. (2020). Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11525-11533).

[18] Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., & Alahi, A. (2018). Social gan: Socially acceptable trajectories with generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2255-2264).

[19] Phan-Minh, T., Grigore, E. C., Boulton, F. A., Beijbom, O., & Wolff, E. M. (2020). Covernet: Multimodal behavior prediction using trajectory sets. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14074-14083).

[20] Zhang, L., Su, P. H., Hoang, J., Haynes, G. C., & Marchetti-Bowick, M. (2020). Map-adaptive goal-based trajectory prediction. arXiv preprint arXiv:2009.04450.

[21] Ngiam, J., Caine, B., Vasudevan, V., Zhang, Z., Chiang, H. T. L., Ling, J., ... & Shlens, J. (2021). Scene transformer: A unified multi-task model for behavior prediction and planning. arXiv e-prints, arXiv-2106.

[22] Casas, S., Luo, W., & Urtasun, R. (2018, October). Intentnet: Learning to predict intention from raw sensor data. In Conference on Robot Learning (pp. 947-956). PMLR.

[23] Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., ... & Hays, J. (2021). Argoverse 2: Next Generation Datasets for Self-Driving Perception and Forecasting.

[24] Ettinger, S., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., ... & Anguelov, D. (2021). Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 9710-9719).

[25] Zhou, Zikang and Ye, Luyao and Wang, Jianping and Wu, Kui and Lu Kejie. (2022). HiVT: Hierarchical Vector Transformer for Multi-Agent Motion prediction. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR2022).

[26] Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H., & Chandraker, M. (2017). Desire: Distant future prediction in dynamic scenes with interacting agents. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 336-345).

[27] Varadarajan, B., Hefny, A., Srivastava, A., Refaat, K. S., Nayakanti, N., Cornman, A., ... & Sapp, B. (2021). MultiPath++: Efficient Information Fusion and Trajectory Aggregation for Behavior Prediction. arXiv preprint arXiv:2111.14973.

[28] Ye, M., Xu, J., Xu, X., Cao, T., & Chen, Q. (2022). DCMS: Motion Forecasting with Dual Consistency and Multi-Pseudo-Target Supervision. arXiv preprint arXiv:2204.05859.