Importance is in your attention: agent importance prediction for autonomous driving

Christopher Hazard, Akshay Bhagat, Balarama Raju Buddharaju, Zhongtao Liu,
Yunming Shao, Lu Lu, Sammy Omari, Henggang Cui
Motional

Abstract

Trajectory prediction is an important task in autonomous driving. State-of-the-art trajectory prediction models often use attention mechanisms to model the interaction between agents. In this paper, we show that the attention information from such models can also be used to measure the importance of each agent with respect to the ego vehicle's future planned trajectory. Our experiment results on the nuPlans dataset show that our method can effectively find and rank surrounding agents by their impact on the ego's plan.

1. Introduction

In order to navigate in the dynamic environment, the autonomous vehicle needs to detect the current locations of the other agents in the environment and predict their future trajectories. Start-of-the-art trajectory prediction models use deep neural networks with attention mechanisms [9, 10] to model the interactions between agents [2–7]. Those prediction models also often include the ego vehicle in the interaction graph in order to model the interactions between the other agents and the ego vehicle.

In addition to the predicted future trajectories, a downstream motion planning module consuming these predictions can also benefit from knowing how much another agent is likely to affect the future maneuvering of the ego vehicle [8]. For example, an agent that is currently behind and traveling away from the ego is not likely to have much impact on the ego's plan while a vehicle making a lane change in front of the ego is very significant. With this knowledge, the motion planner module can focus its computational resources on handling the more important agents and potentially use coarser-grained methods to handle agents with low importance.

The focus of this work is to predict the importance score of the other agents. The most straightforward approach is to model the importance prediction task as a classification problem and train a prediction head using supervised training. However, this approach requires the ground-truth labels for the agent importance scores, and such labels can be hard to obtain.

In this paper, we propose a simple method to predict the importance score of the other agents without requiring any extra training labels that takes advantage of the fact that most of the state-of-the-art trajectory prediction models already use a built-in attention mechanism to model the interactions between agents in a graph. Through a series of experiments, we show that the attention weights between the ego vehicle and other agents can naturally represent the degree to which the existence of each other agent affects the predicted maneuvering of the ego vehicle.

2. Related work

Agent importance prediction is often used in autonomous driving stacks [8], typically calculated with the use of human labeled ground-truth importance scores. Not only are these expensive to label, but label quality is hard to control due to the subjective nature of this task. To address this limitation, [8] proposes to generate the ground-truth labels by running an existing planner in simulation on the dataset and labeling the agent importance based on the planner cost. Our method, on the other hand, does not require any ground-truth labels to train.

The trajectory prediction task predicts the future trajectories of a set of agents, given their history tracks and map information. Since the behavior of an agent also depends on the state of the other agents, it is important for the trajectory prediction model to be able to model the interactions between agents when making predictions. The graph attention mechanism [9, 10] is the most popular approach for modeling such agent interactions.

LaneGCN [5] proposes an Agent-to-Agent attention module to model the agent interactions, and it is also later used by GOHOME [3] and THOMAS [4]. Given the agent input features $\{x_i\}_{i=1}^N$, the Agent-to-Agent attention module computes the agent output features $\{y_i\}_{i=1}^N$ as:

$$y_i = x_i W_0 + \sum_j \phi(\text{concat}(x_i, \Delta_{i,j}, x_j)W_1)W_2 \quad (1)$$

VectorNet [2], SceneTransformer [7], and mmTransformer [6] uses the Transformer attention module proposed in [9]. Given the agent input features $\{x_i\}_{i=1}^N$, Transformer attention computes the agent output features query $\{y_i\}_{i=1}^N$ using Q, key K, and value V matrices:

$$\mathbf{y} = \text{softmax}(\frac{Q(\mathbf{x})K(\mathbf{x})^T}{\sqrt{d}})V(\mathbf{x}) \quad (2)$$

In order to model the interactions between the other agents and the ego vehicle, those prediction models also often include the ego vehicle in the interaction graph and predict the future trajectory of the ego vehicle in the same way as the other agents.

3. Agent importance prediction

3.1. Problem setup

We propose computing the importance score of the agents by using their attention weights with respect to the ego vehicle from the agent to agent interaction module of the trajectory prediction model. The main inputs of this module are the feature vectors of all N actors in the scene, denoted as $\{x_i\}_{i=1}^N$. The outputs of this module are the output feature vectors with the actor interactions modeled, denoted as $\{y_i\}_{i=1}^N$.

$$y_i = \text{Interaction}(\{x_j\}_{j=1}^N) \quad (3)$$

The goal of our work is to predict the importance scores γ from an agent (with feature x_a) to the ego vehicle (with feature x_e) using a pretrained `Interaction` module.

3.2. Attention in a single attention

When there is a single attention layer in the `Interaction` module, it usually has the property that the contributions from each agent j to agent i are computed with a function g and then summed together.

$$y_i = f(x_i) + \frac{\sum_j g(x_i, x_j)}{\text{Normalizer}} \quad (4)$$

With this formulation, we propose to define $g(x_i, x_j)$ as the *attention vector* from agent j to agent i, and use its L2 norm as the importance score from agent j to agent i, which represents the magnitude of agent j's influence on the future trajectory predictions of agent i.

$$\gamma(x_i, x_j) = \|g(x_i, x_j)\|_2 \quad (5)$$

This formulation generalizes the attention modules used by most of the state-of-the-art trajectory prediction works, including the LaneGCN attention module used in LaneGCN [5], GOHOME [3], and THOMAS [4], and the Transformer attention module used in VectorNet and SceneTransformer.

For LaneGCN attention (Eq 1), the attention vector is simply

$$g(x_i, x_j) = \phi(\text{concat}(x_i, \Delta_{i,j}, x_j)W_1)W_2 \quad (6)$$

The Transformer attention (Eq 2) has the softmax operation, but we can expand its formula as

$$\begin{aligned} y_i &= \sum_j \frac{e^{Q(x_i)K(x_j)^T}}{\sqrt{d}\sum_k e^{Q(x_i)K(x_k)^T}} V(x_j) \\ &= \frac{\sum_j e^{Q(x_i)K(x_j)^T)} V(x_j)}{\sqrt{d}\sum_j e^{Q(x_i)K(x_j)^T}} \end{aligned} \quad (7)$$

Which gives us

$$g(x_i, x_j) = e^{Q(x_i)K(x_j)^T)} V(x_j) \quad (8)$$

3.3. Attention in multiple layers

When there are multiple attention layers in the `Interaction` module, we can compute the agent importance scores on each of the attention layers and aggregate them. Our evaluation result in Section 4.4 shows that we get similar performance by taking the average importance score, maximum importance score, or just the importance score from the last attention layer.

4. Evaluation

4.1. Experiment setup

We built our agent importance prediction module on top of the LaneGCN [5] model, which shares the same Agent-to-Agent attention module as GOHOME [3] and THOMAS [4]. We trained the model on the nuPlan [1] training dataset and ran the trained model on 2000 randomly selected validation samples. The prediction horizon is 8 seconds. The LaneGCN model contains two attention layers for Agent-to-Agent attention, and by default, we computed the agent importance scores from the last attention layer. Since we use agent attention value to compute the agent importance score, we will use "attention value" and "importance score" interchangeably in this section.

4.2. Correlation between agent attention and ego behavior change

In this set of experiments, we show that the agents with high predicted importance scores are indeed the ones that have high impacts on the ego behavior. To demonstrate this, we sort the agents in each scene by their importance score, remove each of them, and measure how the predicted ego trajectory will change. [1]

[1] Here we use the ego trajectory predicted from the prediction model as a proxy to the ego plan from the motion planner.

We show the results in Table 1. We calculate the correlation between the normalized attention value (i.e., importance score) assigned to the removed agent and the pointwise L2 distance of the predicted ego trajectory before and after removing the agent, as well as the correlation to the amount of change in the prediction L2 error (w.r.t. ground-truth ego trajectory) before and after removal. We report both Pearson correlation and R-squared values for each quantity. The R-squared values correspond to the amount of variance explained by the dependent variable in a linear model.

k-th Agent	**Pred traj delta corr.** **Pearson**	 R^2	**Pred error delta corr.** **Pearson**	 R^2
1	.477	.128	.228	.016
2	.341	.073	.116	.005
3	.211	.028	.044	.0008
All	.200	.117	.040	.014

Table 1. We remove the k-th highest attended agent, and show the correlation between the predicted ego trajectory delta and attention value, as well as the correlation between the predicted ego trajectory error (w.r.t. ground-truth ego trajectory) delta and attention value. We report both Pearson correlation and R-squared values for each quantity. The last row contains the results for an experiment in which we remove all other agents in the scene, and we compute the correlation using the sum of the attention values of all agents.

Quantity	**0**	**30**	**50**	**80**	**90**	**100**
Highest attns	.003	.117	.165	.245	.307	.748
# relevent agents	0	1	2	3	3	7
Max attn shift	.003	.306	.371	.485	.553	.812
Traj anglar delta [rad]	0	.001	.002	.01	.044	.283

Table 2. Quantile distributions.

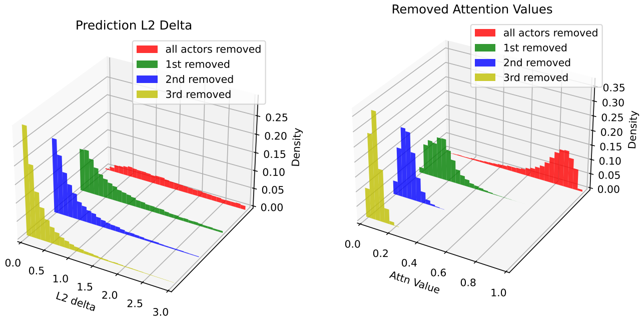

Figure 1. Distribution of ego trajectory L2 delta between pre- and post- removal of the highest attended agent and the amount of attention removed in each experiment. We note that the plot of L2 distances has its domain scaled between (0,3) meters to show the bulk of the density, however there are a significant number of outliers going out to as far as 20 meters in total error, meaning that our distributions have very long tails.

The correlation between the attention value of the agent removed and the change in predicted ego trajectory is highly positive, indicating that our method indeed assigns high importance scores to agents that have high impacts on the ego plan. We also observe that the correlation decreases as k gets bigger, meaning the ego trajectory is more correlated with the higher attended agents.

The last row of Table 1, in which we remove all other attended agents, shows a reduction in the Pearson correlation with regards to each of the other rows despite the fact that this experiment removes the most attention. The fact that the Pearson correlation is so small indicates that overall the total removed attention was less effective than the same amount of attention attributed to the other single removal experiments. Given that the Pearson correlations decrease with regards to rank, we can conclude that our model is rightfully assigning more attention to the most important agents as desired.

Figure 1 shows the distribution of ego trajectory delta between pre- and post- removal of the highest attended agent. It also confirms the conclusion that agents that are predicted to have high importance scores (i.e., high attention values) also have high importance on the ego behavior.

In Table 2, we show the quantile distribution of 1) attention value of the highest attended agent, 2) the number of agents with at least 0.1 normalized attention value, 3) maximum (among all agents) attention changes between the two attention layers, and 4) trajectory angular delta. The trajectory angle is defined as the angle of the vector from the ego's current position to the last predicted waypoint, and we compute the delta between the pre- and post- agent removal predictions. We find the vast majority of ego vehicles experience very small and typically negligible maximum angle change. However, in some rare cases, there are significant angular deltas, such as when the highest likelihood modality changes at an upcoming fork in the road depending on the agent in front of the ego, or more commonly when our prediction switches to an adjacent lane change.

4.3. Spatial attention distribution

In Figure 2, we plot the average normalized attention value (i.e. importance score) per 4×4 meter block in the ego-centric coordinate frame. We can see that the majority of the attention is placed on agents directly in front of our ego agent since those are the agents whose future behavior would most likely cause the ego agent to behave differently. In comparison, very little attention is typically paid to vehicles behind the ego since their presence usually wouldn't impact how the ego vehicle should behave unless the ego needed to adjust its path to avoid a future collision with the agent behind it.

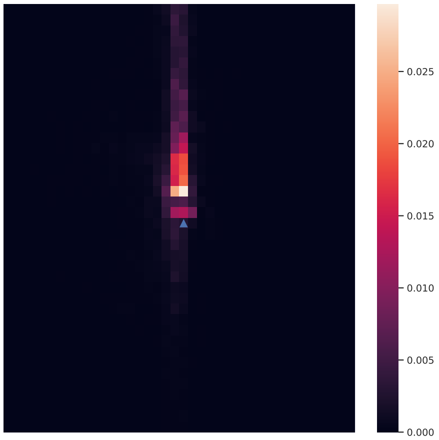

Figure 2. Normalized attention per 4 × 4 meter block in the ego-centric coordinate frame. The ego position and orientation is marked with the blue triangle (in the middle and pointing north).

4.4. Attention in multiple layers

	Pred traj delta corr.	
Aggre Function	Pearson	R^2
Max	.524	.275
Mean	.498	.248
Last	.512	.262

Table 3. Correlation between ego trajectory delta and attention values with different attention aggregation functions.

In this experiment, we perform an ablation study to compare different aggregation functions to aggregate the attention values (i.e., importance scores) from multiple attention layers. In Table 3, we compared using three aggregation functions: 1) maximum attention among all layers, 2) average attention among all layers, and 3) only use the attention of the last layer. We compared the correlation between ego trajectory delta and the importance score, and the result shows that all three functions yield similar results, which means our method is robust to the selection of aggregation function.

5. Conclusion and future works

In this work, we have studied the allocation of attention to agents surrounding our ego vehicle and shown that the normalized magnitude of the attention vector produced by the model for each agent is a good indicator of the underlying importance of each agent. By using those attention values as the agent importance scores, we are able to properly prioritize agents that will have high impacts on the ego trajectory.

In our evaluation we used the ego trajectory predicted from the prediction model as a proxy for the ego plan from a motion planner: in future work we will integrate this importance prediction module into a real motion planner.

References

[1] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 2

[2] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020. 1, 2

[3] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Gohome: Graph-oriented heatmap output forfuture motion estimation. *arXiv e-prints*, pages arXiv–2109, 2021. 1, 2

[4] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Thomas: Trajectory heatmap output with learned multi-agent sampling. *arXiv preprint arXiv:2110.06607*, 2021. 1, 2

[5] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European Conference on Computer Vision*, pages 541–556. Springer, 2020. 1, 2

[6] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7577–7586, 2021. 1, 2

[7] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified multi-task model for behavior prediction and planning. *arXiv e-prints*, pages arXiv–2106, 2021. 1, 2

[8] Khaled S Refaat, Kai Ding, Natalia Ponomareva, and Stéphane Ross. Agent prioritization for autonomous navigation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2060–2067. IEEE, 2019. 1

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 2

[10] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 1