Supplementary Material for "ADAPT: Efficient Multi-Agent Trajectory Prediction with Adaptation"

Abstract

In this supplementary, we present the details of implementation (Section 1) and training (Section 2), perform a complexity analysis of our method compared to other methods (Section 3), provide an additional ablation study on the model components (Section 4), identify failure cases of our method (Section 5), and finally visualize additional qualitative results (Section 6). We also visualize the result of multiple attention heads in Section 6 specializing in different areas and agents in the scene, which might help future work on interpretability.

1. Implementation Details

Scene Representation: We follow VectorNet [2] in our polyline subgraph implementation to obtain the updated node features $\mathbf{v}_i$ of the subgraph as follows:

$$\mathbf{v}_i^{(l+1)} = \text{cat}\left(\text{MLP}_l(\mathbf{v}_i^{(l)}),\ \text{pool}(\text{MLP}_l(\mathbf{v}^{(l)}))\right)$$

where $\mathbf{v}_i^{(l)}$ denotes the feature vector of agent i at layer l, pool the max-pool operation, and cat the concatenation operation. We use an MLP with 2 linear layers and ReLU for non-linearity with a layer normalization [1] after the first layer. We set the layer number l to 3 and the size of the feature vector to 128 for both the agent and the lane subgraph.

Interaction Modelling: For each multi-head attention block (MHAB), we set the number of attention heads to 8 and apply a dropout rate of 0.1 to the attention probabilities. We set the size of the hidden layer in the feed-forward networks to 128 and the number of iterations L to 3.

Meta Info: Meta info includes the location of the agent at time t, $t-1$, and the yaw angle at t. Locations are in 2D coordinates and the angle is in radians, resulting in a 5-dimensional vector. We concatenate the meta info to the corresponding agent feature before decoding.

Trajectory Predictor: For both dynamic and static heads, we use a 2-layer MLP with ReLU for the non-linearity and a layer normalization [1] after the first layer. Differently from subgraphs, we use residual connections in the last layer.

2. Training Details

As mentioned in the paper, we use the variety loss to capture multi-modal futures by calculating the loss only for the most accurate trajectory over K predicted ones. Given the ground truth trajectory $\{\mathbf{s}_t\}_{t=1}^{T}$ and the predicted trajectory with the closest endpoint $\{\hat{\mathbf{s}}_t\}_{t=1}^{T}$ for T future steps, we train our model using the endpoint loss $\mathcal{L}_{end}$, the full trajectory loss $\mathcal{L}_{traj}$, and the trajectory classification loss $\mathcal{L}_{cls}$. $\mathcal{L}_{end}$ is the difference between the closest endpoint and the ground truth endpoint:

$$\mathcal{L}_{end} = \mathcal{L}_{\text{Smooth-}\ell_1}(\hat{\mathbf{s}}_T,\ \mathbf{s}_T) \tag{1}$$

where $\hat{\mathbf{s}}_T$ is the endpoint of $\hat{\mathbf{s}}$, i.e. the prediction at time T. $\mathcal{L}_{traj}$ is the mean of the per-step difference between the predicted full trajectory, $\hat{\mathbf{s}}$, and the ground truth trajectory, $\mathbf{s}$:

$$\mathcal{L}_{traj} = \frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{\text{Smooth-}\ell_1}(\hat{\mathbf{s}}_t,\ \mathbf{s}_t) \tag{2}$$

Finally, $\mathcal{L}_{cls}$ is the Binary Cross Entropy Loss applied to the assigned probabilities $\mathbf{p}$ of K trajectories where the ground truth probability of the closest trajectory $\hat{\mathbf{s}}$ is set to 1 and the others to 0:

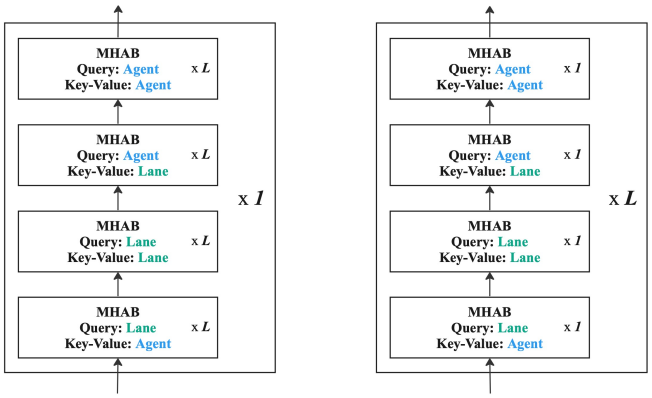
$$\mathcal{L}_{cls} = \mathcal{L}_{\text{BCE}}(\mathbf{p},\ \mathbf{y}) \tag{3}$$

where $\mathbf{y}$ denotes the ground truth probabilities assigned. Overall, our loss is the sum of these three losses:

$$\mathcal{L} = \mathcal{L}_{end} + \mathcal{L}_{traj} + \mathcal{L}_{cls} \tag{4}$$

3. Computational Complexity

In this section, we provide a comparison of the computational complexity according to the attention operations used in the existing approaches. We first define variables that define the number of elements. N, M, and T correspond to the number of agents, lane elements, and time steps, respectively. T can be decomposed into two variables, T_p and T_f, which refer to past and future time steps, respectively. In general, the number of agents dominates the computation, then, the number of lanes followed by the fixed number of time steps, e.g. $T = 50$ ($N > M > T$). While the number



(a) Sequential Order (b) Iterative Order

Figure 1: **The Order of Attention in Interaction.** Given a number of layers $\boldsymbol{L}$, there are two ways of applying attention to model interactions: sequential and iterative. In sequential order a, each type of interaction is considered $\boldsymbol{L}$ times sequentially. In iterative order b, each type of interaction is considered once in a single pass and the pass is repeated $\boldsymbol{L}$ times.

of lanes M stays mostly uniform across scenes, the number of agents N might vary significantly even for the same scene.

As addressed in the SceneTransformer [6], directly applying attention to both time and agent axes results in high overhead, with the computational complexity of $\mathcal{O}((NT + M)^2)$ where N is the number of agents, M is the number of lane segments and T is the number of time steps including both past and future. SceneTransformer reduces it to $\mathcal{O}(NT^2 + N^2T + NTM)$ with factorized attention over time and agent axes.

Autobot [3] does not include lane elements in their factorized attention steps. Contrary to SceneTransformer, their encoding and decoding phases consider only past and future time steps, respectively, resulting in the complexity of $\mathcal{O}(NT_p{}^2 + N^2T_p + NT_f{}^2 + N^2T_f)$ where T_p denotes the number of past time steps and T_f denotes the number of future time steps.

HiVT [9] does not use the standard multi-axis factorized attention but embraces a more efficient type of temporal interaction by considering only one agent for each time step and attending to only one feature over different time steps. Since HiVT follows an agent-centric approach and calculates agent features independently from each other, considering only one agent in their local scene does not result in information loss. However, the agent-centric approach comes with the overhead of N runs of the same procedure. Considering scene normalization for each agent and global interaction in the end, HiVT has the overall complexity of $\mathcal{O}(N^2T_p + NT_p{}^2 + NM)$.

ADAPT has a clear advantage in terms of computational complexity over the existing approaches. Our computation is not bounded by T as our subgraphs in vectorized encoder handle the temporal reasoning. Since we calculate the attention over only agents and lanes, ADAPT has the complexity of $\mathcal{O}(N^2 + NM + M^2)$ resulting from the attention operations in the interaction modeling. Removing the number of time steps T out of the equation is the main reason behind the efficiency gain of ADAPT.

4. Quantitative Results

In this section, we present an additional ablation study to justify some minor design choices. Specifically, we investigated the effect of iterative vs. sequential order in interaction (Fig. 1) and the effect of using two separate subgraphs for encoding agents and lanes. The results in Table 1 show that the iterative attention blocks outperform their sequential counterpart. This implies that updating intermediate features at each iteration, as opposed to the attention order used in LaneGCN [5], leads to a better understanding of the relationship between agents and lanes. Furthermore, the use of separate polyline subgraphs for lanes and agents, which is in contrast to prior work [4, 2], produces better results. Overall, our decision choices on the architecture improve performance with better feature encoding.

5. Failure Cases

In this section, we provide some failure cases and investigate possible reasons. We perform the analysis on single-agent predictions on Argoverse, since the miss rate in single-agent prediction is relatively higher than multi-agent predictions on Interaction. We identify three sources of error for failure cases: erroneous data, missing rare behaviors, and inaccurate predictions.

Erroneous Data: The accuracy of the provided input trajectories in the past directly affects the future predictions, since the future predictions are trained to be consistent with the past ones. Thus, defective or unstable history data causes incorrect future predictions as shown in Fig. 2a.

	mADE_6	mFDE_6	MR_6
w/o Iterative Att.	0.673	0.971	0.086
w/o Dual Subgraph	0.671	0.960	0.086
ADAPT	**0.668**	**0.948**	**0.083**

Table 1: **Single-Agent Ablation Study on Argoverse (Val.).** This table shows the effect of iterative attention and dual subgraph on the performance of single-agent prediction on the Argoverse validation set.

Moreover, defects in the future steps result in inaccurate evaluations of the predictions (Fig. 2b). Some problems such as id-switch and position-oscillation, resulting in unstable and incorrect ground truth future locations, are addressed in previous works as well [8, 7].

Additionally, accurate map information plays an important role in future predictions because it directly affects the reasoning of the model about drivable areas. In the example shown in Fig. 2c, predictions are intensified on a single mode i.e. left turn, because of the missing lane that the ground truth trajectory follows.

Missing a Peculiar Mode: Despite the large number of scenarios on Argoverse, some behaviors are less frequently observed such as a u-turn or an abrupt lane change. These behaviors that are rare on the training set cause the model to miss the relevant mode at test time as shown in Fig. 2d.

Precision of Predictions: Some predictions result in an error due to a lack of precision in the predicted trajectories despite correctly identifying intention. For example, in Fig. 2e, all possible paths are covered by the predictions but the difference between the closest endpoint and the ground truth endpoint is higher than the miss rate threshold.

6. Qualitative Results

In this section, we provide additional qualitative results for both single-agent (Fig. 3) and multi-agent (Fig. 4) predictions of ADAPT on Argoverse and Interaction validation sets, respectively.

Focus of Attention Heads: In Fig. 5, we visualize attention scores from different MHAB heads that are used to update an agent (red) in scenes from the validation split of the Interaction dataset. The attention scores are gathered from **AA** and **LA** modules for agents and lanes, respectively. In the first layer of interaction (the first row), attention heads do not focus on any specific scene elements yet as this is the first step where the agent is informed by the scene. As the agent has no prior information, attending to all scene elements without focusing on any is a reasonable choice in the first layer. On the other hand, in the next layer, each attention head specializes in some part of the map. For example, in the scene given in the upper set of rows, head 2 attends to lanes in the upper left of the map whereas head 4 attends to the upper right lanes. Some **AA** heads attend only to the agent itself and do not consider any other agents, e.g. head 3 and head 4. In the last layer, heads still attend to some specific parts of the map and a subset of the agents. These visualizations confirm that different types of interactions are captured with the multi-head attention blocks of our model.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv.org*, 2016. 1

[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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[3] Roger Girgis, Florian Golemo, Felipe Codevilla, Martin Weiss, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2022. 2

[4] Junru Gu, Chen Sun, and Hang Zhao. DenseTNT: End-to-end trajectory prediction from dense goal sets. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 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 *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 2

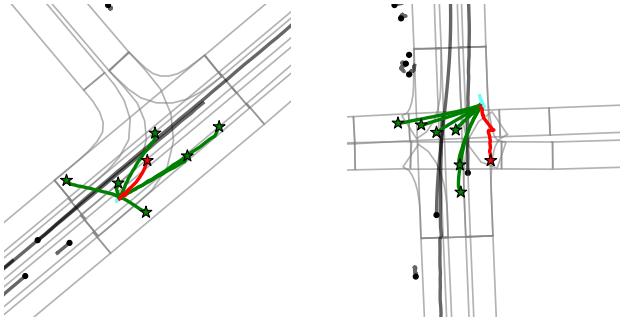
[6] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2022. 2

[7] Haoran Song, Di Luan, Wenchao Ding, Michael Y Wang, and Qifeng Chen. Learning to predict vehicle trajectories with model-based planning. In *Proc. Conf. on Robot Learning (CoRL)*, 2021. 3

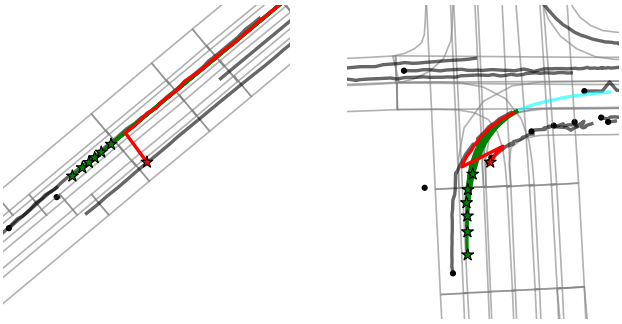
[8] Maosheng Ye, Tongyi Cao, and Qifeng Chen. TPCN: Temporal point cloud networks for motion forecasting. In *Proc. IEEE*

Conf. on Computer Vision and Pattern Recognition (CVPR), 2021. 3

[9] Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

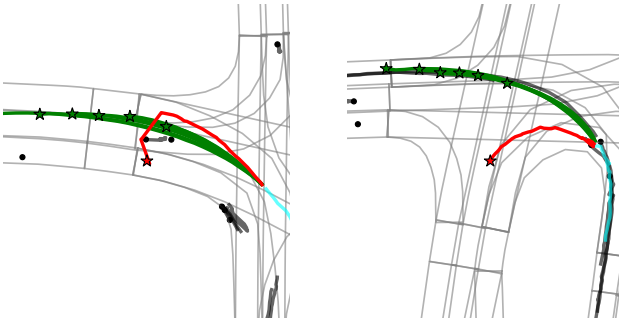


(a) **Erroneous Input Trajectories.** The errors in the given trajectories, both past, and future, result in unreasonable futures due to inconsistent and uninformative past locations.

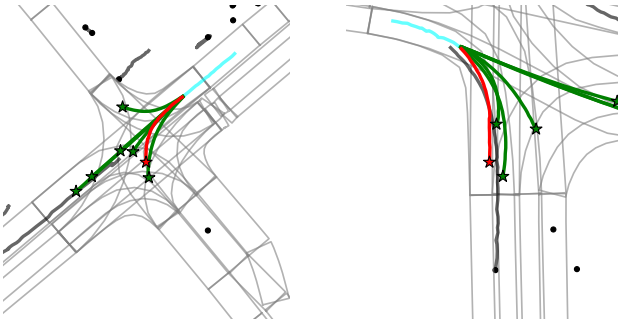


(b) **Erroneous Ground Truth**. Erroneous (impossible) ground truth evaluates admissible trajectories as failures.

(c) **Problems in the Input Map.** The missing lane information on the map causes the model to miss a possible future path.



(d) **Missing a Mode.** Rare behaviors cause the model to miss the mode corresponding to the ground truth.



(e) **Inaccurate Predictions**. The difference between the predicted and the ground truth endpoints is higher than the miss rate threshold, therefore this case is classified as a failure although ground truth intention is correctly captured by the predictions.

Figure 2: **Failure Cases on Argoverse**. We present some failure cases with their potential reasons.

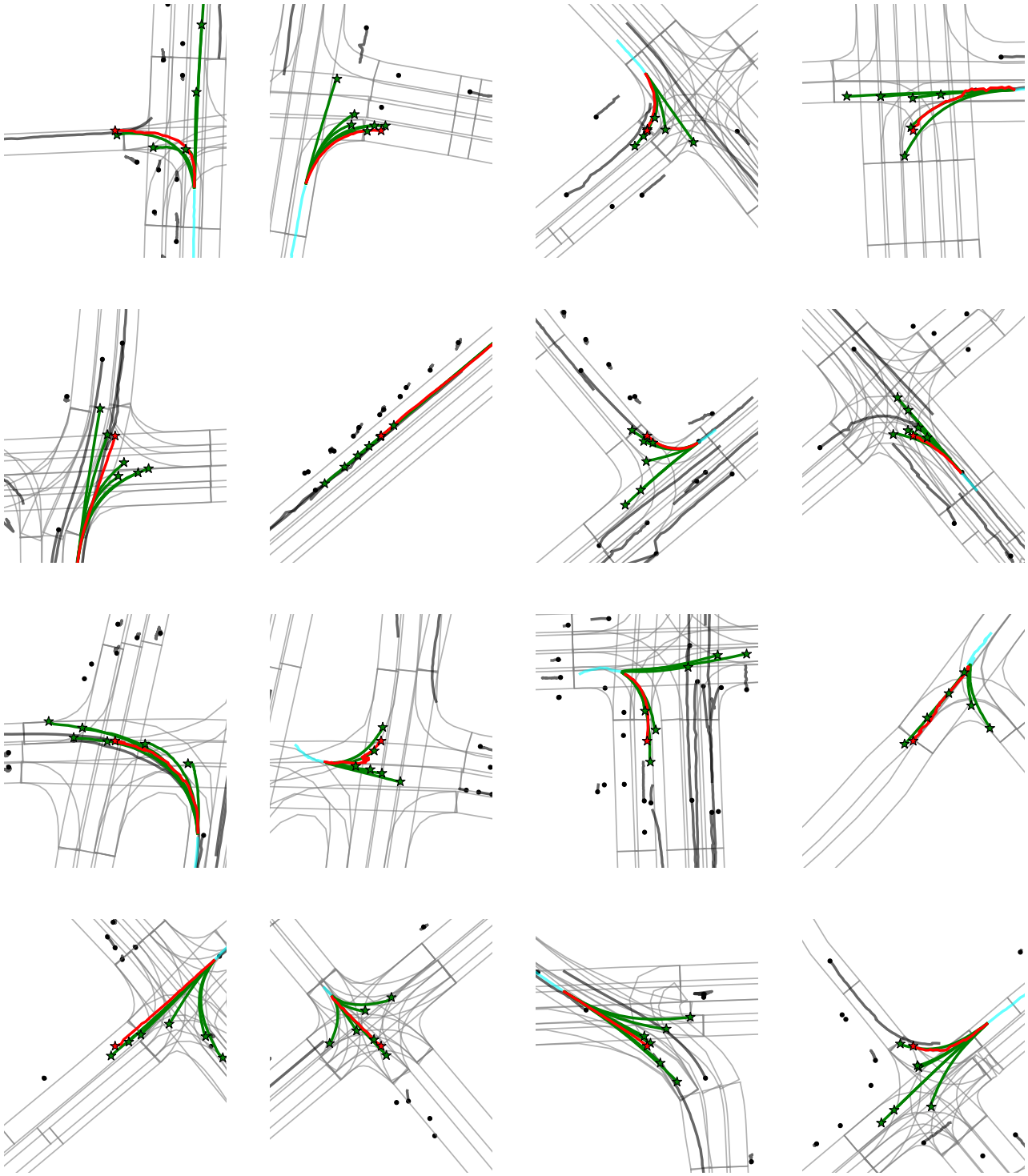


Figure 3: **Additional Qualitative Results for Single-Agent Predictions on Argoverse.** The red colored trajectory shows the ground truth future, the cyan shows the past trajectory of the agent of interest, and the green trajectories are the multiple predictions. Context agents are displayed in black. ADAPT can successfully predict a trajectory similar to the ground truth by also covering possible diverse trajectories for agent of interest.

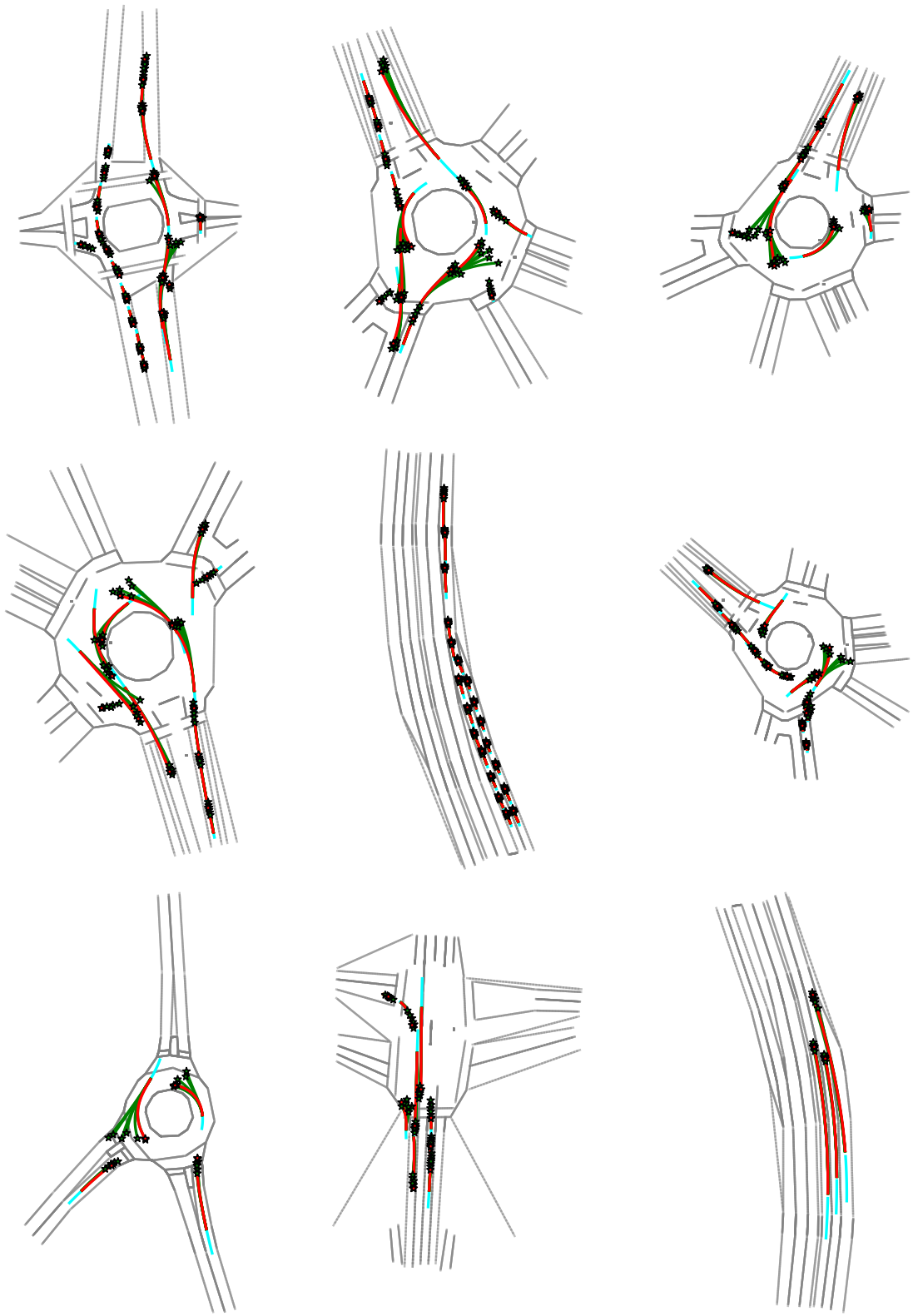


Figure 4: **Additional Qualitative Results for Multi-Agent Predictions on Interaction.** The red colored trajectories show the ground truth future for each agent, the cyan shows the past trajectories of the agents, and the green trajectories are the predictions. ADAPT can successfully predict futures for each agent in a single forward pass without introducing additional overhead.

Figure 5: **Visualization of Attention Scores on the Interaction**. We visualize multi-head attention from different layers for a selected agent (red). The attention probabilities for the agents (blue) and lanes (green) are the results of the Agent-Agent and the Lane-Agent modules, respectively. The transparency increases with lower attention probabilities. As the attention propagates towards higher layers, the attention heads specialize towards specific components such as lanes in the right turn, the vehicle in front, etc.