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

Scene Informer: Anchor-based Occlusion Inference and Trajectory Prediction in Partially Observable Environments

Bernard Lange[1], Jiachen Li[2], and Mykel J. Kochenderfer[1]

***Abstract*— Navigating complex and dynamic environments requires autonomous vehicles (AVs) to reason about both visible and occluded regions. This involves predicting the future motion of observed agents, inferring occluded ones, and modeling their interactions based on vectorized scene representations of the partially observable environment. However, prior work on occlusion inference and trajectory prediction have developed in isolation, with the former based on simplified rasterized methods and the latter assuming full environment observability. We introduce the Scene Informer, a unified approach for predicting both observed agent trajectories and inferring occlusions in a partially observable setting. It uses a transformer to aggregate various input modalities and facilitate selective queries on occlusions that might intersect with the AV's planned path. The framework estimates occupancy probabilities and likely trajectories for occlusions, as well as forecast motion for observed agents. We explore common observability assumptions in both domains and their performance impact. Our approach outperforms existing methods in both occupancy prediction and trajectory prediction in partially observable setting on the Waymo Open Motion Dataset. Our implementation with additional visualizations is available at https://github.com/sisl/SceneInformer.**

I. INTRODUCTION

Safe navigation through dynamic environments necessitates reasoning about both occluded and visible parts of the environment. Traffic participants analyze the social cues from other agents and the topography of the scene to infer hypothetical future scenarios. For instance, a vehicle slowing down near a crosswalk could imply a pedestrian is about to cross. Experienced drivers incorporate the uncertainty tied to potential states of the occluded environment into their decision-making to enhance maneuver safety. Similar reasoning is crucial in safety-critical and dynamic robotic applications, such as self-driving cars. This paper aims to tackle the task of occlusion inference and trajectory prediction in autonomous vehicles (AVs).

In the context of AVs, a suite of sensors delivers an environment representation to the perception and sensor fusion modules. These modules generate representations used in downstream tasks such as trajectory prediction and planning. Naturally, these sensors are sometimes partially occluded, leading to incomplete representations and erroneous outcomes from the downstream tasks. Hence, occlusion awareness is critical to safe driving. Two primary methods can integrate occlusion reasoning into an AV: occlusion-aware decision making [1]–[6], which involves planning conservatively around occluded areas, and occlusion inference [7]–[9] with a focus on deducing the presence of objects in the occlusions. This work is focused on occlusion inference, due to its ability to offer an explicit representation of inferred occluded agents for downstream tasks, and its integration with trajectory prediction of observed agents.

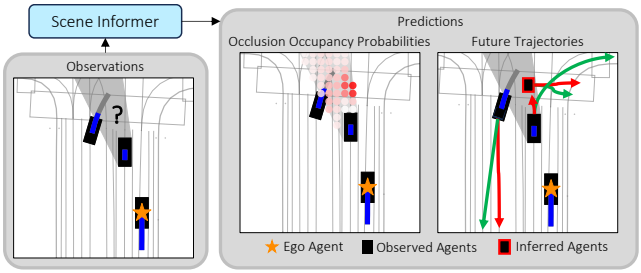

Fig. 1: We introduce Scene Informer, an end-to-end prediction framework that considers both observed and occluded agents in a partially observable environment. It forecasts multi-modal futures for observed agents and estimates occupancy probabilities and most likely trajectories originating from the occlusion.

Effective reasoning about a partially observable environment necessitates considering the observed and occluded agents, their future motion, and their interactions with one another. This involves processing vectorized inputs from perception frameworks, such as agent trajectories, agent properties, and environmental maps. Despite these requirements, occlusion inference and trajectory prediction have each been developed in isolation, with limited integration between the two domains. Prior work on occlusion inference has primarily operated under simplified settings. They reason in terms of fixed-size grids with occupancy probabilities of agents and do not leverage the diverse input modalities available to modern AV [6]–[9]. Furthermore, they do not model the interaction between predicted occluded and visible agents. On the other hand, trajectory prediction approaches that predict the motion of observed agents typically assume full observability of the environment [10]–[13].

To rectify these limitations, we introduce an end-to-end, learning-based framework for environment prediction in partially observable settings called *Scene Informer*. Our proposed method is designed to infer any occlusion of interest and predict the motion of observed agents. It uses a transformer to aggregate an arbitrary number of input modalities that include histories and properties of observed agents along with a lane graph. It allows selective inference of occluded areas, enabling queries for occlusions that might interact with the AV's planned trajectory. It infers a set of oc-

[1]Stanford Intelligent Systems Laboratory, Stanford University, Stanford, CA 94305, USA {blange, jiachen_li, mykel}@stanford.edu
[2]Trustworthy Autonomous Systems Laboratory, University of California, Riverside, Riverside, CA 92507, USA jiachen.li@ucr.edu

cupancy probabilities along with the most likely trajectories for each occlusion, and future trajectories for the observed agents, as shown in Fig. 1. Our framework is the first end-to-end comprehensive environment prediction solution that both infers occlusions and predicts observed agent trajectories.

We evaluate our framework on the Waymo Open Motion Dataset (WOMD) [14] by simulating occlusions with a line-of-sight method from the ego vehicle perspective. Our method is compared with recent work on occlusion inference [7], [8], fully observable trajectory prediction, and simplified variations of Scene Informer. We investigate various observability assumptions prevalent in both occlusion inference and trajectory prediction frameworks, and assess the impact of end-to-end training on overall performance. Our proposed framework achieves state-of-the-art performance in terms of both occupancy prediction and trajectory prediction of occluded objects, and demonstrates increased robustness to partial observability when forecasting observed agents. In summary, our contributions are as follows:

- We propose a novel occlusion inference framework called Scene Informer that uses a transformer to infer selected obstructed parts of the scene, and predict future trajectories of the observed agents.
- We investigate the influence of different observability assumptions commonly used in the occlusion inference and trajectory prediction frameworks and their impact on the final performance.
- We demonstrate the superior performance of our approach on the Waymo Open Motion Dataset (WOMD) compared to prior work on occlusion inference and fully observable trajectory prediction.

II. RELATED WORK

We discuss the prior work on occlusion inference, occlusion-aware decision making, and trajectory prediction.

Occlusion Inference. Prior work uses interactions between observed agents in the scene to infer the obstructed parts of the environment. Afolabi *et al.* [7] clusters observed interactions between human drivers and crosswalks to learn occupancy grid maps (OGMs) of the occlusions. Hara *et al.* [15] uses camera observations to infer the state of the blindspot. Itkina *et al.* [8] accounts for the multimodality of the scene by learning a driver sensor model with a conditional variational autoencoder (CVAE) [16] which maps observed trajectories to a latent vector. Then they fuse multiple maps acquired from different observed agents with evidential theory [17] to generate the OGM of the occluded area. Christianos *et al.* [6] extends it by proposing a two-stage training procedure with an additional CVAE that predicts future trajectories of inferred agents. Prior work has relied on fixed-size OGMs and rasterized networks and is not trained end-to-end. They fail to capture the semantics of the scenes and attempt to infer all occlusions. We propose a single-stage architecture that uses vectorized representations to infer any occlusion in terms of occupancy and likely future trajectories, and forecast trajectories of observed agents.

Occlusion Aware Decision Making. The planner can incorporate sensor occlusions in the observation model [1], [5], or include imaginary agents in the occlusions often following some worst-case scenario analysis [4], [18]–[20]. Bouton *et al.* [1] demonstrates that a POMDP policy can safely navigate with sensor occlusions. Chae *et al.* [18] incorporates imaginary occluded objects into the overtaking controller. Wray *et al.* [4] includes virtual agents outside the vehicle's field of view in the observation model of a T-intersection POMDP policy. Hoermann *et al.* [19] aggregates both object-based and object-free representations (e.g. occupancy grid maps) to safely navigate left turns. Hubmann *et al.* [20] generalizes phantom vehicle reasoning to any urban scenarios. Nager *et al.* [21] extends it to different classes of virtual agents. Mun *et al.* [5] encodes occluded observations with a variational model [22] and trains an agent to navigate crowded scenes. Christianos *et al.* [6] predicts likely occluded trajectories for the planner.

Fully Observable Trajectory Prediction. Trajectory prediction forecasts the motion of agents based on the assumption of full observability. The effectiveness of the approach is determined by the methodology used to aggregate the input modalities. In rasterized approaches [23]–[28], all modalities are represented with image-like representations and encoded with convolutional network. An alternative way is to represent different input modalities as a set and process them with recurrent neural networks [29]–[31] or graph neural networks [12], [32]–[39]. Recently, transformer-based approaches [13], [40], [41] have been achieving state-of-the-art results on numerous benchmarks. In our proposed approach, we adopt a transformer-based framework to infer occlusions and predict the motion of observed agents addressing the unrealistic full observability assumption of prior methods.

III. SCENE INFORMER

Reasoning over observed and occluded agents that might interfere with a planned trajectory is critical to ensure the safety of the AV maneuvers. It involves processing a range of vectorized input modalities provided by upstream perception frameworks, such as observed agents and lane graphs while taking into account the partial observability of the environment. We propose an environment prediction framework called *Scene Informer* (see Fig. 2) that takes in observed trajectories of other agents, vectorized maps, and infers the true state of the occlusion interest and the future motion of observed agents. Our framework consists of transformer encoder and decoder pairing. The encoder takes in vectorized representation of the observed agents and maps to create the scene embeddings [40], [41]. The decoder receives the scene embeddings and reasons in terms of spatial anchors for both observed agents and occlusions. For observed agents, spatial anchors indicate the most recent position. In the case of occlusions, they represent points randomly placed within the occlusion of interest. It allows intelligent queries for occlusions that might interact with the planned trajectory rather than inferring all occlusions at once, which is computationally expensive and unnecessary.

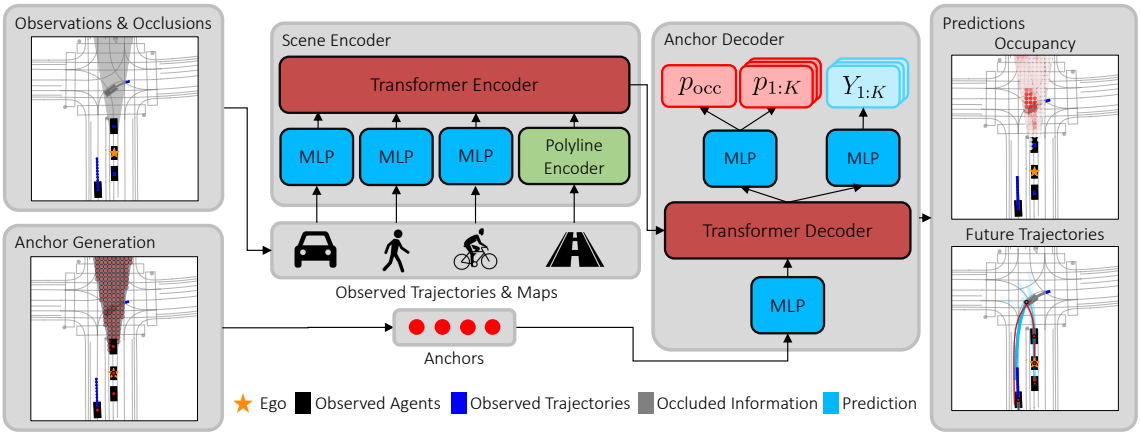

Fig. 2: Scene Informer consists of a scene encoder and anchor decoder. It reasons in terms of anchors (●) that are assigned to each observed agent and randomly populated in the occlusion of interest. Scene encoder aggregates different observation input modalities and creates scene embeddings. Anchor decoder cross-attends between scene embeddings and anchors, and outputs predicted occupancy probability p_{occ} and K most likely future trajectories $Y_{1:K}$ (■) with the probability of each trajectory $p_{1:K}$ for each anchor.

Finally, for each provided anchor, the decoder infers a probability of occupancy and the k most likely trajectories with their corresponding probabilities to model the multi-future behavior of the observed and potentially present but occluded agents.

A. Problem Statement and Input Representation

Given an environment state representation S_{past} that consists of observed agent information O_{past} and road map information M, our objective is to infer the occlusion and predict future trajectories from the perspective of the ego vehicle. Agent information $O_{\text{past}} \in \mathbb{R}^{N_o \times H \times D_a}$ represents N_o agents over history horizon H with D_a features consisting of position, heading, velocity, and agent's dimensions. Road map information is denoted as $M \in \mathbb{R}^{N_m \times N_p \times D_m}$ which describes N_m polylines with N_p points with feature vector D_m. Each point is represented by its position, vector to the next point in a polyline, and polyline type (e.g. crosswalk, lane boundary, stop sign, etc.) We represent the occlusion to infer and agents to predict by a set of spatial anchors $A \in \mathbb{R}^{N_a \times 2}$, where N_a is the number of anchors defined by their position. N_o anchors are assigned to the most recent positions of the observed agents, while $N_a - N_o$ anchors are uniformly distributed within the occluded areas, as shown in Fig. 2. For each anchor, we predict k most likely trajectories Y over the prediction horizon P with probability distribution $p_{1:K}$, and probability of the anchor being occupied p_{occ}.

B. Framework

We use a transformer architecture that has been shown to excel at modeling multimodal and unstructured inputs of varying sizes [13], [40], [41]. Our proposed framework consists of *scene encoder* and *anchor decoder*. The scene encoder processes multimodal input, such as past agent trajectories and vectorized maps, and generates scene embeddings. The anchors that define agent and occlusions to predict are provided to the anchor decoder which cross-attends them with the scene embeddings to generate anchor embeddings. Those embeddings are then used to predict the probability of the anchor being occupied, and top k future trajectories with their corresponding probability distribution.

Scene encoder: It aggregates the state representation S_{past} that includes agent information O_{past} and road map information M modalities. Each agent type is encoded with a dedicated multi-layer perceptron, and each road polyline is modeled with a PointNet polyline encoder layer [13], [42] as it allows us to encode polylines of varying lengths:

$$M_{\text{enc}},\ O_{\text{enc}} = \phi(\text{MLP}(M)),\ \text{MLP}(O_{\text{past}}) \tag{1}$$

where $\text{MLP}(\cdot)$ is a multi-layer perceptron network, and ϕ is a max-pooling operator over the last feature dimension. For different agent types, we have a dedicated set of parameters for MLP. Subsequently, all representations are concatenated to form $S_{\text{in}} = [O_{\text{enc}}, M_{\text{enc}}] \in \mathbb{R}^{(N_o+N_m)\times D_{\text{enc}}}$. They are provided to the transformer encoder which aggregates input modalities with a varying number of tokens:

$$S_{\text{emb}} = \text{TransformerEncoder}(S_{\text{in}}) \tag{2}$$

where $S_{\text{emb}} \in \mathbb{R}^{(N_o+N_m)\times D_{\text{out}}}$ are scene embeddings.

Anchor decoder: We define anchors $A \in \mathbb{R}^{N_a \times 2}$. Each anchor is projected to a feature dimension D_{enc} consistent with other encodings. It is then provided to the transformer decoder which cross-attends them with other anchors and scene embeddings from the scene encoder:

$$A_{\text{emb}} = \text{TransformerDecoder}(\text{MLP}(A), S_{\text{emb}}) \tag{3}$$

where $A_{\text{emb}} \in \mathbb{R}^{N_a \times D}$ are anchor embeddings. The embedding of each anchor is provided to MLP to predict the probability of occupancy p_{occ}, K most likely future trajectories represented with Gaussian Mixture Model $Y \in \mathbb{R}^{K \times P \times 5}$, and a probability distribution over trajectories $p_{1:K}$. Each time step of the predicted trajectory Y is represented with parameters $(\mu_x, \mu_y, \sigma_x, \sigma_y, \rho)$ that define a Gaussian component [13].

Training loss: Our loss is a mix of regression, trajectory classification, and anchor occupancy classification components. For a given anchor i, if it is occupied, we identify the predicted mode j out of K possible trajectories with a hard-assignments strategy [10], [43]. Subsequently, we maximize

the log-likelihood associated with the selection of mode j, the generation of the ground truth trajectory GT by mode j, and the occupation of anchor i. Conversely, if anchor i remains unoccupied, the log-likelihood is maximized for its unoccupied state. The cumulative loss is computed by aggregating these components over all anchors:

$$\max \mathbb{1}[i = \text{occ.}] \left(\log \Pr(j \mid A_{\text{emb},i}) + \log \Pr(GT \mid Y_{i,j}) \right) + \log \Pr(i \mid A_{\text{emb},i}). \quad (4)$$

IV. EXPERIMENTS

A. Dataset

We evaluate our framework on the Waymo Open Motion Dataset (WOMD) [14]. It contains approximately 103k 20s-long scenes recorded at 10Hz and consists of object tracks and vectorized maps. To our best knowledge, it is the only large open-source autonomous driving dataset that contains labels sourced from an offboard perception module. The offboard method is a non-causal approach that is not constrained by real-time performance and uses future observations to detect past objects [44]. As a result, it offers unparalleled accuracy in tracking objects within occlusions. We simulate occlusions with a line-of-sight method in the bird's-eye view (BEV) of an ego-vehicle. While occlusions occur in three dimensions, we investigate the integration of partial observability reasoning into the environment prediction framework which operates in a BEV representation. In our experiments, we explore various degrees of observability:

- *Full observability*: The true state is fully observed, commonly used in trajectory prediction approaches.
- *Limited observability*: Only agents visible via line-of-sight from the ego vehicle's perspective are observed.
- *Partial observability*: Agents visible from the ego vehicle's perspective are observed, along with other randomly selected agents that might be occluded.

The true AV's observability lies between full and limited observability. A suite of sensors combined with tracking systems can often detect what is behind obstructions and track agents through temporary occlusions. Nevertheless, an AV inherently lacks access to the true state of the environment, which underscores the importance of environment prediction in a partially observable setting.

Each training sample consists of vectorized representations, such as 1 s of history, 4 s of ground truth future, polyline-based map representation, and a occlusion created by a single agent. Temporal information is discretized with 0.1 s step. We reduce the dimensionality of the road map by sampling points separated by at least 1.5 meters and limiting the observable region to 60 meters around the ego vehicle. Anchors are assigned to the last time step of observed agents and randomly populated within the occlusion polygon. An example of a sample is shown in Fig. 3.

B. Implementation details

Scene encoder: We implement three MLPs for cars, bikes, pedestrians, and a polyline encoder for maps. Each output a set of feature vectors with a size of 256 for each observed agent and each polyline. We use a transformer encoder with four layers, four heads, 256 hidden size, and a 2048 feedforward dimension.

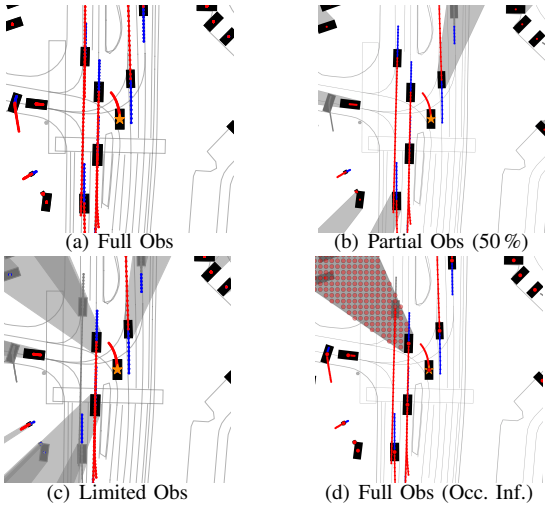

Fig. 3: Visualization of the dataset. Each sample contains agents (■) with a history of observations (■) and future trajectory (■) in the frame of the ego vehicle (★). We explore the following variations: (a) Full Observability. (b) Partial Observability with 50 % of generating occlusions (■). (c) Limited Observability with all possible occlusions. (d) Full Observability with a single occlusion (used for Scene Informer training). Anchors (●) are assigned to the last time step of observed agents and populated in the occlusion. Occluded agents and observations are grey (■).

Anchor decoder: It consists of an MLP that encodes anchors, a two-layer transformer with other parameters the same as the encoder, and output MLP heads for occupancy classification, trajectory classification, and trajectory prediction. We consider seven future trajectory modes.

Training details: The number of trainable parameters is 11.3M. We train each model with the AdamW [45] optimizer with a linear learning rate warmup from 0.0 to 0.0001 over the first 10k gradient steps. All models are implemented in PyTorch [46], and trained with Lightning AI 2.0.6 [47] in mixed precision with a batch size of 20 accumulated over 2 steps for 10 epochs (250k gradient steps). Experiments were carried out on an NVIDIA TITAN RTX 24GB GPU with an AMD Ryzen 3960X CPU and 64 GB of RAM.

C. Experimental Setup and Metrics

We assess our framework's performance in terms of occlusion inference of unobserved agents and trajectory prediction of observed agents in a partially observable environment. Our evaluation metrics encompass the classification of occupancy for unobserved agents and a regression analysis comparing predicted trajectories to the ground truth for observed and unobserved agents. In the classification task, we assess the effectiveness in inferring which anchors are occupied within occlusions, focusing on the classification accuracy for occupied and free anchors (ACC_{OCC}/ACC_{FREE}). For the regression task, we evaluate the alignment between our predicted trajectories and the ground truth using minimum

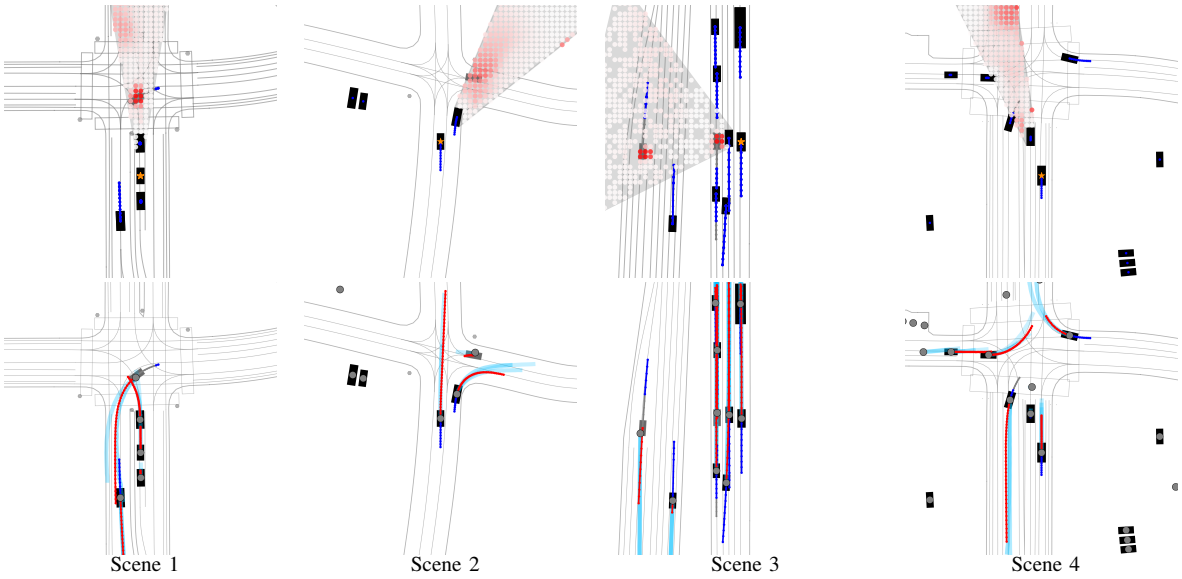

Fig. 4: Scene Informer Predictions in Crowded Settings: The ■ visualizes predictions, its intensity reflecting trajectory probability. The ● intensity signifies occupancy probability. The top row displays occupancy, the bottom, forecasted trajectories. The ● denotes trajectories from high-likelihood occupancy anchors. Our method reliably predicts significant occupancy for ground truth occluded objects, delivering realistic trajectories for all agents. In scenes 1-3, our model accurately gauges occluded agent positions and their future paths. In scene 4, with the ego (★) nearing a crosswalk with stationary vehicles, our approach anticipates a crossing pedestrian.

average and final displacement errors (ADE_{min}/FDE_{min}).

We evaluate the performance on occluded agents by benchmarking against existing occlusion inference methods, such as K-means PaS [7], GMM PaS [8], and MAVOI [8]. None of the open-source approaches is capable of inferring future trajectories of occluded agents. To evaluate the inferred trajectory, we compare our method with variations of our framework: the vanilla trajectory prediction that reasons only about observed agents, and an occlusion inference adaptation that focuses exclusively on reasoning over occlusions. Even though the vanilla trajectory prediction assumes full observability, we can still query it to infer the trajectory of occluded agents by providing an anchor with the ground truth position of the occluded agent. For the observed agents, we compare Scene Informer with a vanilla trajectory prediction variations of our framework trained on fully observable data and limited observability data, respectively. The latter lets us determine whether we can enhance the robustness of vanilla trajectory prediction to partial observability without explicitly incorporating occlusion inference reasoning. We then assess the robustness of all models by evaluating them under various observability scenarios. This involves using a dataset where the likelihood of an occlusion being generated by an agent varies. Contrary to the training dataset, it applies to all agents potentially creating multiple occlusions in the scene. It ranges from 0 % (full obs.), with increments of 25 %, 50 %, and 75 % (partial obs.), to 100 % (limited obs.) (see Fig. 3).

V. Discussion

Occlusion inference performance: We report the evaluation of the occlusion inference performance of our approach compared to the baselines and the impact of evaluation observability assumptions in Table I. Our method consistently outperforms all baselines in both occupied and free accuracy metrics, regardless of the visibility setting. The improvement is in the range of 10.5 %-27.5 % and 17.0 %-35.0 % points on occupied and free cells, respectively. Notably, as the visibility assumption in the evaluation dataset gets more restrictive, our model tends to predict more occupied cells—a potentially desirable trait for safety-critical applications.

On the regression task, Scene Informer outperforms a vanilla trajectory prediction baseline and a variation trained only on the occluded agents. As the observation becomes more limited, the performance gap between models, particularly when compared to vanilla trajectory prediction, widens in favour of our approach. In the limited observability setting, the FDE_{min} of Scene Informer is 2.37 m less than that of the vanilla trajectory prediction, which underscores the ability of our framework to robustly reason about occlusions across a range of observability scenarios. In addition, our framework outperforms its variation trained solely on occluded agents, which implies that occlusion inference and trajectory prediction are complementary tasks.

Figure 4 provides examples of inferred occlusions. As shown, our approach effectively predicts occupancy and the future trajectory based on other agents' motions and road layouts. The predicted occupancy map realistically reflects the true motion of partially observable traffic, infers potential unobserved agents and assigns a higher probability of occupancy to actively used parts of drivable spaces. Figure 5 shows how the observed motion of the traffic participants impacts the occlusion inference. We modify the observations to indicate potentially different flow of traffic. We demon-

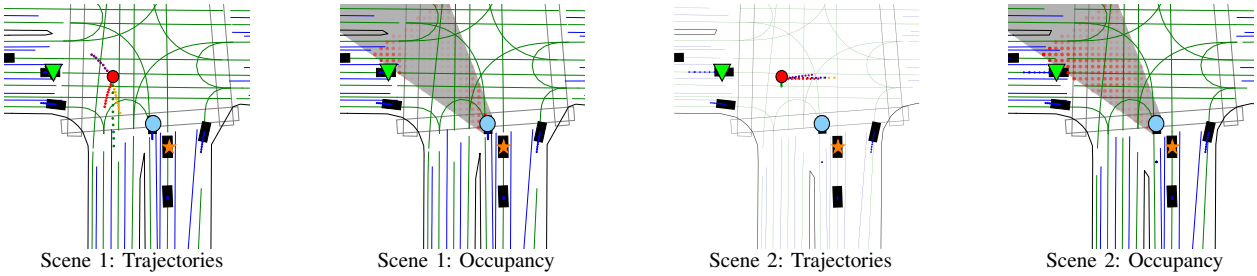

Fig. 5: Impact of observed agents' histories on the occlusion inference performance. We modify the observed trajectories of two agents (●and ▼). In Scene 1, we visualize a scenario where a ▼ is stationary and a ● is moving forward. For an anchor ●, our approach predicts occupancy with low probability and vertical motion from the top to the bottom of the intersection. In Scene 2, we visualize the predictions for a modified scenario. The ▼ is approaching quickly the intersection and the ● is stationary. Scene Informer realistically adapts its occupancy and trajectory predictions based on the observed motion of other agents. It is now highly likely that the anchor in the middle of the occlusion is occupied, and a majority of predicted trajectories are horizontal from the left to right of the intersection.

TABLE I: Comparison on occlusion inference and trajectory prediction in terms of classification accuracy and displacement errors. For occluded agents, "Full" denotes "Full Obs (Occ. Inf.)".

Model	Full 0%	25%	Partial 50%	75%	Limited 100%
Occlusion Classification Accuracy ($ACC_{OCC}(\uparrow)$ / $ACC_{FREE}(\uparrow)$)					
K-means PaS [7]	65.7 / 40.6	65.5 / 41.3	65.8 / 41.0	66.1 / 40.8	66.5 / 40.7
GMM PaS [8]	61.3 / 47.5	58.1 / 50.4	56.0 / 51.6	54.4 / 52.8	53.0 / 53.9
MAVOI [8]	67.9 / 58.6	65.5 / 55.7	67.1 / 56.1	67.3 / 55.4	66.7 / 55.0
Ours	**78.4 / 75.6**	**79.8 / 74.0**	**80.4 / 73.4**	**80.7 / 72.6**	**80.5 / 72.8**
Occluded Agents' Trajectory Prediction ($ADE_{min}(\downarrow)$/$FDE_{min}(\downarrow)$)					
Traj. Pred.	1.94 / 2.97	2.08 / 3.22	2.23 / 3.49	2.37 / 3.76	2.50 / 4.00
Occlusion Inf. Only	1.15 / 1.96	1.19 / 2.02	2.08 / 1.23	1.26 / 2.13	1.30 / 2.19
Ours	**0.87 / 1.43**	**0.91 / 1.50**	**0.95 / 1.54**	**0.98 / 1.59**	**1.00 / 1.63**
Observed Agents' Trajectory Prediction ($ADE_{min}(\downarrow)$/$FDE_{min}(\downarrow)$)					
Traj. Pred. (Full Obs)	**0.26 / 0.62**	0.35 / 0.80	0.45 / 0.97	0.53 / 1.12	0.60 / 1.26
Traj. Pred. (Limited)	0.28 / 0.67	0.33 / 0.78	0.38 / 0.88	0.42 / 0.97	0.45 / 1.04
Ours (Full Obs)	**0.26 / 0.62**	**0.31 / 0.73**	**0.36 / 0.83**	**0.40 / 0.91**	**0.43 / 0.99**

strate that Scene Informer adapts its predictions realistically in response to changes in the behavior of observed agents.

Trajectory prediction performance: Table I compares the performance of our framework in predicting the trajectories of observed agents only and compares it with the commonly used vanilla trajectory prediction method. In a fully observable evaluation, our framework aligns with the performance of a vanilla trajectory prediction approach, as expected. However, as the evaluation accounts for increasing partial observability, our method consistently surpasses vanilla trajectory predictions, with the performance difference also becoming more evident with a higher probability of occlusions (0.27 m in final displacement error in the limited observability). Vanilla trajectory prediction methods assume that all interacting agents are present in the scene representations and do not account for missing or incomplete observations. The introduction of partial observability can then result in a corrupted scene representation, leading to inferior performance. In the real world, this manifests as flickering or missing agents due to physical obstructions or adverse weather conditions. Furthermore, a train-test distribution shift might arise due to differences in observation labeling strategies. While the training set may be manually labeled or automated by an off-board system, detections during deployment are provided by real-time, less accurate ones leading to incomplete observations. It underscores the importance of modeling partial observability and making trajectory prediction frameworks more robust. We compare our approach with the vanilla trajectory prediction method, but trained in a limited observability setting. While it improves performance in the limited setting, it underperforms in the fully observable one. Yet, it still fails to match the performance of Scene Informer, which has been trained on the fully observable adaptation of the dataset. It indicates that incorporating occlusion reasoning enhances robustness in partial observability settings. Our frameworks can reason over agents that might be present but are not observed.

VI. CONCLUSION

We present the first end-to-end comprehensive environment prediction framework that reasons about observable and unobservable parts of an environment. Scene Informer realistically infers any occlusion of interest and predicts future trajectories for observed agents. Unlike prior work in occlusion inference, it is an end-to-end framework and is not restricted to a fixed-size occupancy grid map. Moreover, it challenges the unrealistic full observability assumption in trajectory prediction. Our results demonstrate that Scene Informer surpasses existing occlusion inference methods, provides robustness in trajectory prediction to partial observability, and underscores the advantages of merging occlusion inference with trajectory prediction. In future work, we aim to incorporate the interactions between occlusions, predict multiple possible occupancy probabilities, and explore its integration with planning. Many trajectory prediction models do not rely on any raw sensor data. This might cause them to overlook critical details [41], [48], leading to cascading errors [49]. They also focus on trajectory predictions rather than joint scene predictions, which are critical for planning.

ACKNOWLEDGMENT

This project was made possible by funding from the Ford-Stanford Alliance.

REFERENCES

[1] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Scalable decision making with sensor occlusions for autonomous driving," in *International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2076–2081.

[2] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 586–595.

[3] X. Lin, J. Zhang, J. Shang, Y. Wang, H. Yu, and X. Zhang, "Decision making through occluded intersections for autonomous driving," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2019, pp. 2449–2455.

[4] K. H. Wray, B. Lange, A. Jamgochian, *et al.*, "Pomdps for safe visibility reasoning in autonomous vehicles," in *IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, 2021, pp. 191–195.

[5] Y.-J. Mun, M. Itkina, S. Liu, and K. Driggs-Campbell, "Occlusion-aware crowd navigation using people as sensors," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 12 031–12 037.

[6] F. Christianos, P. Karkus, B. Ivanovic, S. V. Albrecht, and M. Pavone, "Planning with occluded traffic agents using bi-level variational occlusion models," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 5558–5565.

[7] O. Afolabi, K. Driggs–Campbell, R. Dong, M. J. Kochenderfer, and S. S. Sastry, "People as sensors: Imputing maps from human actions," in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2342–2348.

[8] M. Itkina, Y.-J. Mun, K. Driggs-Campbell, and M. J. Kochenderfer, "Multi-agent variational occlusion inference using people as sensors," in *International Conference on Robotics and Automation (ICRA)*, 2022.

[9] M. Krueger, P. Palmer, C. Diehl, T. Osterburg, and T. Bertram, "Recognition beyond perception: Environmental model completion by reasoning for occluded vehicles," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 999–11 006, 2022.

[10] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," in *Conference on Robot Learning (CoRL)*, 2020.

[11] B. Ivanovic, A. Elhafsi, G. Rosman, A. Gaidon, and M. Pavone, "MATS: An interpretable trajectory forecasting representation for planning and control," in *Conference on Robot Learning (CoRL)*, 2021.

[12] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *International Conference on Computer Vision (ICCV)*, 2021, pp. 15 303–15 312.

[13] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion transformer with global intention localization and local movement refinement," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[14] S. Ettinger, S. Cheng, B. Caine, *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *International Conference on Computer Vision (ICCV)*, 2021, pp. 9710–9719.

[15] K. Hara, H. Kataoka, M. Inaba, K. Narioka, R. Hotta, and Y. Satoh, "Predicting vehicles appearing from blind spots based on pedestrian behaviors," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8.

[16] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.

[17] A. P. Dempster, "A generalization of bayesian inference," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 30, no. 2, pp. 205–232, 1968.

[18] H. Chae and K. Yi, "Virtual target-based overtaking decision, motion planning, and control of autonomous vehicles," *IEEE Access*, vol. 8, pp. 51 363–51 376, 2020.

[19] S. Hoermann, F. Kunz, D. Nuss, S. Renter, and K. Dietmayer, "Entering crossroads with blind corners. a safe strategy for autonomous vehicles," in *Intelligent Vehicles Symposium (IV)*, 2017, pp. 727–732.

[20] C. Hubmann, N. Quetschlich, J. Schulz, J. Bernhard, D. Althoff, and C. Stiller, "A pomdp maneuver planner for occlusions in urban scenarios," in *Intelligent Vehicles Symposium (IV)*, IEEE, 2019, pp. 2172–2179.

[21] Y. Nager, A. Censi, and E. Frazzoli, "What lies in the shadows? safe and computation-aware motion planning for autonomous vehicles using intent-aware dynamic shadow regions," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 5800–5806.

[22] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *International Conference on Learning Representations (ICLR)*, 2014.

[23] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 336–345.

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

[25] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions," in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8454–8462.

[26] H. Cui, V. Radosavljevic, F.-C. Chou, *et al.*, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2090–2096.

[27] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "Covernet: Multimodal behavior prediction using trajectory sets," in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14 074–14 083.

[28] C. Choi, J. H. Choi, J. Li, and S. Malla, "Shared cross-modal trajectory prediction for autonomous driving," in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 244–253.

[29] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 683–700.

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

[31] H. Ma, Y. Sun, J. Li, M. Tomizuka, and C. Choi, "Continual multi-agent interaction behavior prediction with conditional generative memory," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8410–8417, 2021.

[32] S. Casas, C. Gulino, R. Liao, and R. Urtasun, "Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data," in *International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9491–9497.

[33] D. Cao, J. Li, H. Ma, and M. Tomizuka, "Spectral temporal graph neural network for trajectory prediction," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 1839–1845.

[34] S. Khandelwal, W. Qi, J. Singh, A. Hartnett, and D. Ramanan, "What-if motion prediction for autonomous driving," *arXiv preprint arXiv:2008.10587*, 2020.

[35] J. Li, F. Yang, M. Tomizuka, and C. Choi, "Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 19 783–19 794, 2020.

[36] H. Girase, H. Gang, S. Malla, *et al.*, "Loki: Long term and key intentions for trajectory prediction," in *International Conference on Computer Vision (ICCV)*, 2021, pp. 9803–9812.

[37] J. Gao, C. Sun, H. Zhao, *et al.*, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 525–11 533.

[38] J. Li, H. Ma, Z. Zhang, J. Li, and M. Tomizuka, "Spatio-temporal graph dual-attention network for multi-agent prediction and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 556–10 569, 2021.

[39] R. Zhou, H. Zhou, H. Gao, M. Tomizuka, J. Li, and Z. Xu, "Grouptron: Dynamic multi-scale graph convolutional networks for group-aware dense crowd trajectory forecasting," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 805–811.

[40] J. Ngiam, B. Caine, V. Vasudevan, *et al.*, "Scene Transformer: A unified architecture for predicting future trajectories of multiple agents," in *International Conference on Learning Representations (ICLR)*, 2021.

[41] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 2980–2987.

[42] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.

[43] B. Varadarajan, A. Hefny, A. Srivastava, *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.

[44] C. R. Qi, Y. Zhou, M. Najibi, *et al.*, "Offboard 3d object detection from point cloud sequences," in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6134–6144.

[45] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2019.

[46] A. Paszke, S. Gross, F. Massa, *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 8026–8037.

[47] W. Falcon and The PyTorch Lightning team, *PyTorch Lightning*, version 1.4, Mar. 2019.

[48] B. Lange, M. Itkina, and M. J. Kochenderfer, "Lopr: Latent occupancy prediction using generative models," *arXiv preprint arXiv:2210.01249*, 2022.

[49] H. Delecki, M. Itkina, B. Lange, R. Senanayake, and M. J. Kochenderfer, "How do we fail? stress testing perception in autonomous vehicles," in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 5139–5146.