Supplementary Material: HCNAF

Geunseob (GS) Oh
University of Michigan
gsoh@umich.edu

Jean-Sébastien Valois
Uber ATG
jsvalois@uber.com

A. Ablation Study on the POM forecasting experiment with Virtual Simulator Dataset

In this section, we present the results of an ablation study conducted on the test set (10% of the dataset) of the *Virtual Simulator* experiment to investigate the impact of different hyper-networks inputs on the POM forecasting performance. As mentioned in Section 4 and 5.1, each 5-second snippet is divided into a 1-second long history and a 4-second long prediction horizons. All the inputs forming the conditions C used in the ablation study are extracted from the history portion, and are listed below.

1. $X^{REF}_{t=-1:0}$: historical states of the reference car in pixel coordinates,
2. $X^{A_i}_{t=-1:0}$: historical states of the actors excluding the reference car in pixel coordinates and up to 3 closest actors,
3. $X^{SS}_{t=0}$: stop-sign locations in pixel coordinates, and
4. $\Omega \in \{\emptyset, \Omega_{All}\}$: bev images of size $N \times 256 \times 256$. N may include all, or a subset of the following channels: stop signs at $t = 0$, street lanes at $t = 0$, reference car & actors images over a number of time-steps $t \in [-1, 0]s$.

As presented in Table S1, we trained 6 distinct models to output $p(X_t|C)$ for $t = 2s, 4s$. The six models can be grouped into two different sets depending on the Ω that was used. The first group is the models that do not utilize any bev map information, therefore $\Omega = \emptyset$. The second group leverages all bev images $\Omega = \Omega_{All}$. Each group can be divided further, depending on whether a model uses $X^{A_i}_{t=-1:0}$ and $X^{SS}_{t=0}$. The HCNAF model which takes all bev images $\Omega = \Omega_{All}$ from the perception model as the conditions C_5 (see Table 1) excluding the historical states of the actors and stop-signs is denoted by the term *best model*, as it reported the lowest NLL. We use M_i to represent a model that takes C_i as the conditions.

Note that the hyper-network depicted in Figure 5 is used for the training and evaluation, but the components of the hyper-network changes depending on the conditions. We also stress that the two modules of HCNAF (the hyper-network and the conditional AF) were trained jointly. Since the hyper-network is a regular neural-network, it's parameters are updated via back-propagations on the loss function.

As shown in Table S1, the second group ($M_{5:6}$) performs better than the first group ($M_{1:4}$). Interestingly, we observe that the model M_1 performs better than $M_{2:4}$. We suspect that this is due to $M_{2:4}$ using *imperfect* perception information. That is, not all the actors in the scene were detected and some actors are only partially detected; they appeared and disappeared over the time span of 1-second long history. The presence of non-compliant, or abnormal actors may also be a contributing factor. When comparing M_2 and M_3 we see that the historical information of the surrounding actors did not improve performance. In fact, the model that only utilizes X^{A_i} at time $t = 0$ performs better than the one using X^{A_i} across all time-steps. Finally, having the stop-sign locations as part of the conditions is helping, as many snippets covered intersection cases. When comparing M_5 and M_6, we observe that adding the states of actors and stop-signs in pixel coordinates to the conditions did not improve the performance of the network. We suspect that it is mainly due to the same reason that M_1 performs better than M_4.

B. Implementation Details on Toy Gaussian Experiments

For the toy gaussian experiment 1, we used the same number of hidden layers (2), hidden units per hidden layer (64), and batch size (64) across all autoregressive flow models AAF, *NAF*, and HCNAF. For *NAF*, we utilized the conditioner (transformer) with 1 hidden layer and 16 sigmoid units, as suggested in [1]. For HCNAF, we modeled the hyper-network with two multi-layer perceptrons (MLPs) each taking a condition $C \in \mathbb{R}^1$ and outputs $\mathbf{W}$ and $\mathbf{B}$. Each MLP consists of 1 hidden layer, a ReLU activation function. All the other parameters were set identically, including those for the Adam optimizer (the learning rate $5e^{-3}$ decays by a factor of 0.5 every 2,000 iterations with no improvement in validation samples). The NLL values in Table 1 were computed using 10,000 samples.

For the toy gaussian experiment 2, we used 3 hidden lay-

Table S1: Ablation study on *Virtual Simulator*. The evaluation metric is negative log-likelihood. Lower values are better.

Conditions		$\Omega = \emptyset$			$\Omega = \Omega_{All}$		
		C_1	C_2	C_3	C_4	C_5	C_6
NLL	**t = 2s**	-8.519	-8.015	-7.905	-8.238	**-8.943**	-8.507
	t = 4s	-6.493	-6.299	-6.076	-6.432	**-7.075**	-6.839
	$C_1 = X^{REF}_{t-\tau:t}$		$C_3 = X^{REF}_{t-\tau:t} + X^{A_{1:N}}_{t-\tau:t}$		$C_5 = X^{REF}_{t-\tau:t} + \Omega_{All}$ (best model)		
	$C_2 = X^{REF}_{t-\tau:t} + X^{A_{1:N}}_{t}$		$C_4 = X^{REF}_{t-\tau:t} + X^{A_{1:N}}_{t-\tau:t} + X^{SS}_{t}$		$C_6 = X^{REF}_{t-\tau:t} + \Omega_{All} + X^{A_{1:N}}_{t-\tau:t} + X^{SS}_{t}$		

ers, 200 hidden units per hidden layer, and batch size of 4. We modeled the hyper-network the same way we modeled the hyper-network for the toy gaussian experiment 1. The NLL values in Table 2 were computed using 10,000 test samples from the target conditional distributions.

C. Number of Parameters in HCNAF

In this section we discuss the computational costs of HCNAF for different model choices. We denote D and L_F as the flow dimension (the number of autoregressive inputs) and the number of hidden layers in a conditional AF. In case of $L_F = 1$, there exists only 1 hidden layer h_{l_1} between X and Z. We denote H_F as the number of hidden units in each layer per flow dimension of the conditional AF. Note that the outputs of the hyper-network are $\mathbf{W}$ and $\mathbf{B}$. The number of parameters for $\mathbf{W}$ of the conditional AF is $N_W = D^2 H_F(2 + (L_F - 1)H_F)$ and that for $\mathbf{B}$ is $N_B = D(H_F L_F + 1)$.

The number of parameters in HCNAF's hyper-network is largely dependent on the scale of the hyper-network's neural network and is independent of the conditional AF except for the last layer of the hyper-network as it is connected to $\mathbf{W}$ and $\mathbf{B}$. The term $N_{1:L_H-1}$ represents the total number of parameters in the hyper-network up to its $L_H - 1$th layer, where L_H denotes the number of layers in the hyper-network. H_{L_H} is the number of hidden units in the L_Hth (the last) layer of the hyper-network. Finally, the number of parameters for the hyper-network is given by $N_H = N_{1:L_H-1} + H_{L_H}(N_W + N_B)$.

The total number of parameters in HCNAF is therefore a summation of N_W, N_B, and N_H. The dimension grows quadratrically with the dimension of flow D, as well as H_F for $L_F \geq 2$. The key to minimizing the number of parameters is to keep the dimension of the last layer of the hyper-network low. That way, the layers in the hyper-network, except the last layer, are decoupled from the size of the conditional AF. This allows the hyper-network to become large, as shown in the POM forecasting problem where the hyper-network takes a few million dimensional conditions.

D. Conditional Density Estimation on MNIST

The primary use of HCNAF is to model conditional probability distributions $p(x_{1:D}|C)$ when the dimension of C (i.e., inputs to the hyper-network of HCNAF) is large. For example, the POM forecasting task operates on large-dimensional conditions with $D_C > 1$ million and works with small autoregressive inputs $D = 2$. Since the parameters of HCNAF's conditional AF grows quickly as D increases (see Section C), and since the conditions C greatly influence the hyper-parameters of conditional AF module (Equation 4), HCNAF is ill-suited for density estimation tasks with $D >> D_C$. Nonetheless, we decided to run this experiment to verify that HCNAF would compare well with other recent models. Table S2 shows that HCNAF achieves the state-of-art performance for the conditional density estimation.

MNIST is an example where the dimension of autoregressive variables ($D = 784$) is large and much bigger than $D_C = 1$. MNIST images (size 28 by 28) belong to one of the 10 numeral digit classes. While the unconditional density estimation task on MNIST has been widely studied and reported for generative models, the **conditional** density estimation task has rarely been studied. One exception is the study of conditional density estimation tasks presented in [2]. In order to compare the performance of HCNAF on MNIST ($D >> D_C$), we followed the experiment setup from [2]. It includes the dequantization of pixel values and the translation of pixel values to logit space. The objective function is to maximize the joint probability over $X := x_{1:784}$ conditioned on classes $C_i \in \{0, ..., 9\}$ of X as follows.

$$p(x_{1:784}|C_i) = \prod_{d=1}^{784} p(x_d|x_{1:d-1}, C_i). \quad (1)$$

For the evaluation, we computed the test log-likelihood on the joint probability $p(x_{1:784})$ as suggested in [2]. That is, $p(x_{1:784}) = \sum_{i=0}^{9} p(x_{1:784}|C_i)p(C_i)$ with $p(C_i) = 0.1$, which is a uniform prior over the 10 distinct labels. Accordingly, the bits per pixel was converted from the LL in logit space to the bits per pixel as elaborated in [2].

For the HCNAF presented in Table S2, we used $L_F = 1$ and $H_F = 38$ for the conditional AF module. For the hyper-network, we used $L_H = 1$, $H_{H,W} = 10$ for $\mathbf{W}$, $H_{H,B} = 50$ for $\mathbf{B}$, and 1-dimensional label as the condition $C \in \mathbb{R}^1$.

Table S2: Test negative log-likelihood (in nats, logit space) and bits per pixel for the conditional density estimation task on **MNIST**. Lower values are better. Results from models other than HCNAF were found in [2]. HCNAF is the best model among the conditional flow models listed.

Models	**Conditional NLL**	**Bits Per Pixel**
Gaussian	1344.7	1.97
MADE	1361.9	2.00
MADE MoG	1030.3	1.39
Real NVP (5)	1326.3	1.94
Real NVP (10)	1371.3	2.02
MAF (5)	1302.9	1.89
MAF (10)	1316.8	1.92
MAF MoG (5)	1092.3	1.51
HCNAF (ours)	**975.9**	**1.29**

E. Detailed Evaluation Results and Visualization of POMs for PRECOG-Carla Dataset

The evaluation results and POM visualizations are presented in the next few pages.

References

[1] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2083–2092, 2018. 1

[2] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017. 2, 3

[3] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. *arXiv preprint arXiv:1905.01296*, 2019. 4

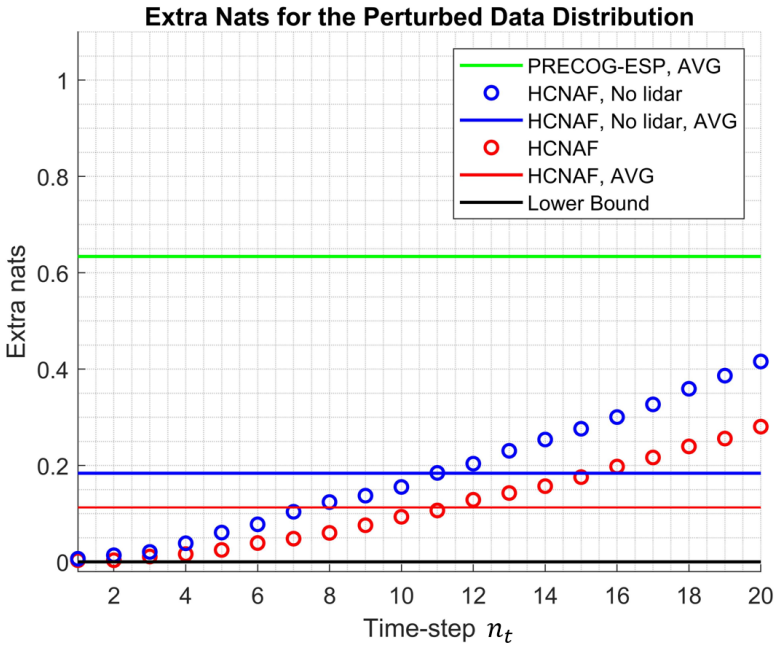

Figure S1: Detailed evaluation results on the *PRECOG-Carla* test set per time-step for the *HCNAF* and *HCNAF(No lidar)* models, compared to average *PRECOG* published performance, such as described in Table 3. AVG in the plot indicates the averaged extra nats of a model over all time-steps $n_{t=1:20}$ (i.e., $\Sigma_{t=1}^{20} \hat{e}_{n_t}/20$). Note that the x-axis time steps are 0.2 seconds apart, thus $n_t = 20$ corresponds to t = 4 seconds into the future and that there is no upper bound of $\hat{e}$ as $\hat{e} \geq 0$. As expected, the POM forecasts $p_{model}(X|C)$ are more accurate (closer to the target distribution $p'(X|C)$) at earlier time-steps, as the uncertainties grow over time. For all time-steps, the HCNAF model with lidar approximates the target distribution better than the HCNAF model without lidar. Both with and without lidar, HCNAF outperforms a state-of-the-art prediction model, *PRECOG-ESP* [3].

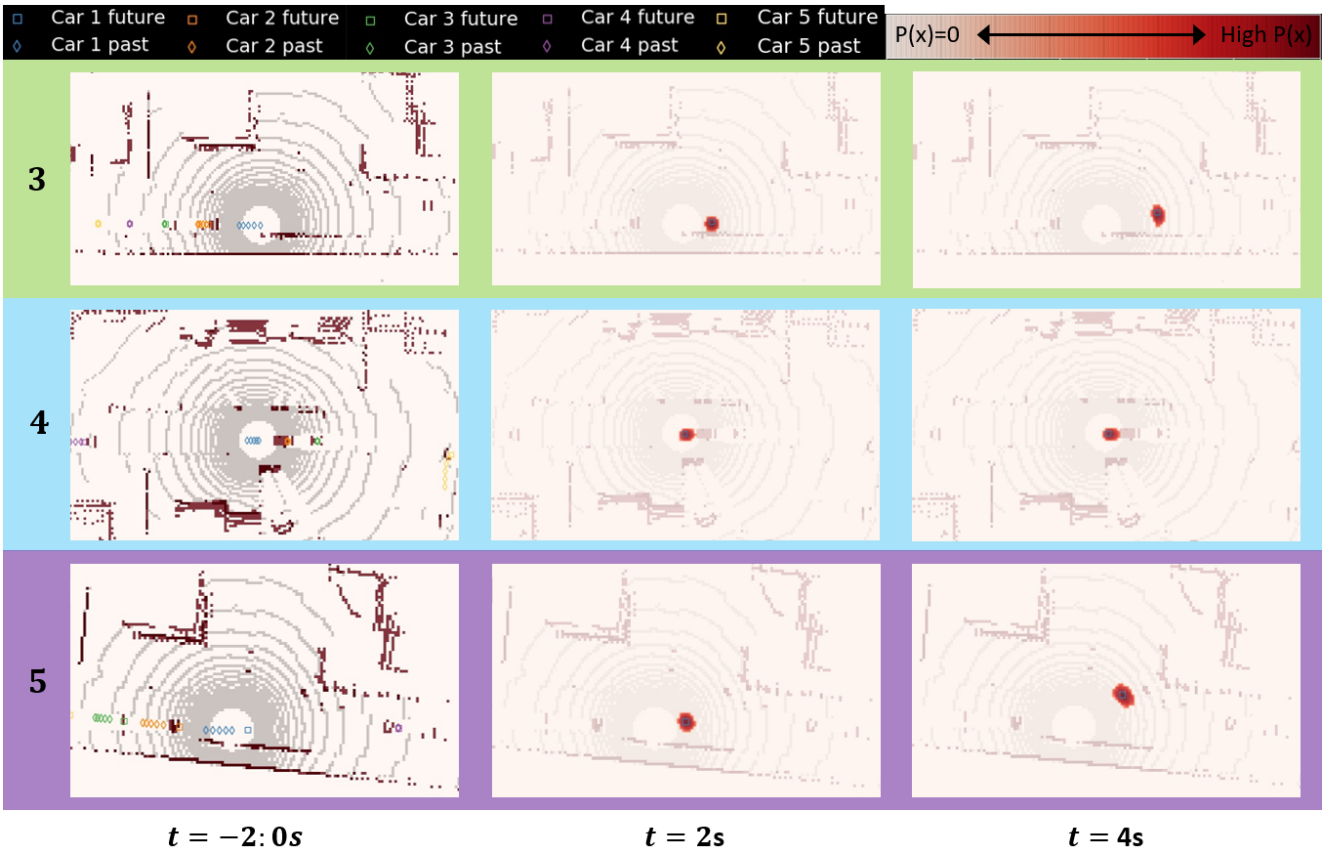

Figure S2: Continuing examples (3 through 5) from the POM forecasts of the HCNAF model described in Table 3 (with lidar) on the *PRECOG-Carla* dataset. In the third example, the car 1 enters a 3-way intersection and our forecasts captures the two natural options (left-turn & straight). Example 4 depicts a 3-way intersection with a queue formed by two other cars in front of car 1. HCNAF uses the interactions coming from the front cars and correctly forecast that car 1 is likely to stop due to other vehicles in front if it. In addition, our model captures possibilities of the queue resolved at $t = 4$s and accordingly predicts occupancy at the tail. The fifth example illustrates car 1 while starting a turn left as it enters the 3-way intersection. The POM forecast for $t = 4$s is an ellipse with a longer lateral axis, which reflects the higher uncertainty in the later position of the car 1 after the turning.

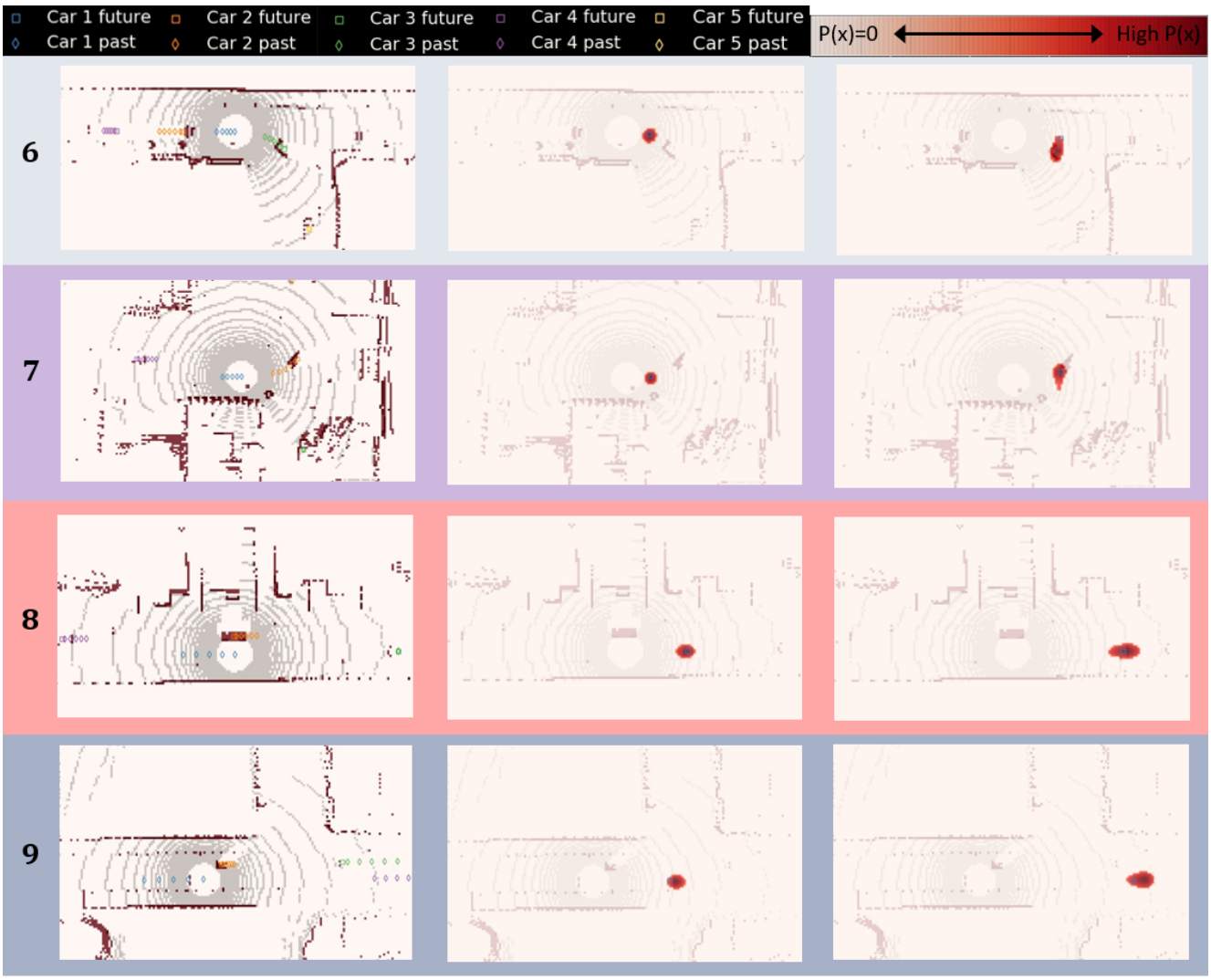

Figure S3: Continuing examples (6 through 9) from the POM forecasts of the HCNAF model described in Table 3 (with lidar) on the *PRECOG-Carla* dataset. In examples 6 and 7, the car 1 enters 3-way intersections and POM shows that HCNAF forecast the multi-modal distribution successfully (straight & right-turn for the example 6, left-turn & right-turn for the example 7). Example 8 depicts a car traveling in high-speed in a stretch of road. The POM forecasts are wider-spread along the longitudinal axis. Finally, example 9 shows a car entering a 4-way intersection at high-speed. HCNAF takes into account the fact that car 1 has been traveling at high-speed and predicts the low likelihood of turning left or right, instead forecasting car 1 to proceed straight through the intersection.

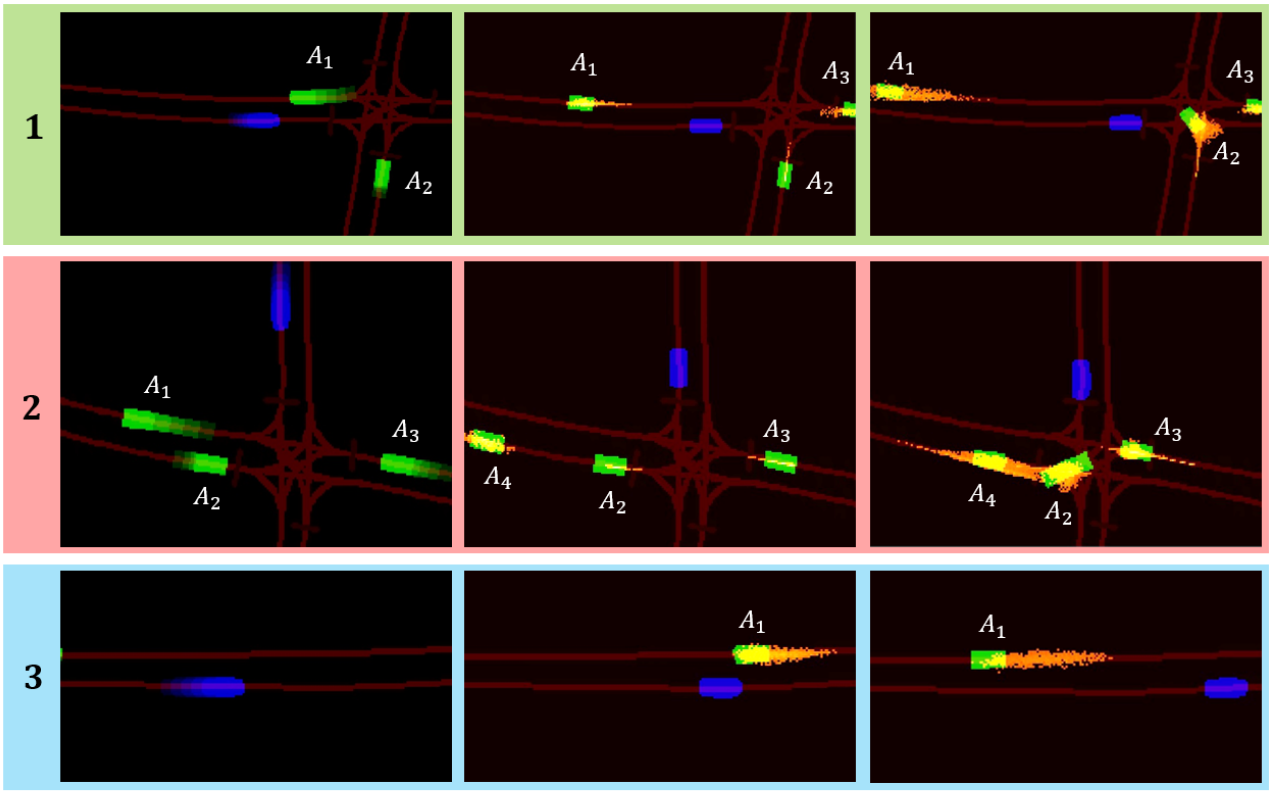

Figure S4: HCNAF for forecasting POMs on our *Virtual Simulator* dataset. Left column: one-second history of actors (green) and reference car (blue). Actors are labeled as A_i. Center and right columns: occupancy prediction for actor centers x_t, y_t, at t = 2 and 4 secs., with actor full body ground truth overlayed. Note that actors may enter and exit the scene. In example 1, our forecasts captured the speed variations of A_1, the stop line deceleration and the multi-modal movements (left/right turns, straight) of A_2, and finally the stop line pausing of A_3. In Example 2, HCNAF predicts A_2 coming to a stop and exiting the intersection before A_3, while A_3 is yielding to A_2. Finally, example 3 shows that HCNAF predicts the speed variations along a stretch of road for A_1.