Raising context awareness in motion forecasting

Hédi Ben-Younes[*,1], Éloi Zablocki[*,1], Mickaël Chen[1], Patrick Pérez[1], Matthieu Cord[1,2]

[1] Valeo.ai, [2] Sorbonne Université

{hedi.ben-younes,eloi.zablocki,mickael.chen,patrick.perez,matthieu.cord}@valeo.com

Abstract

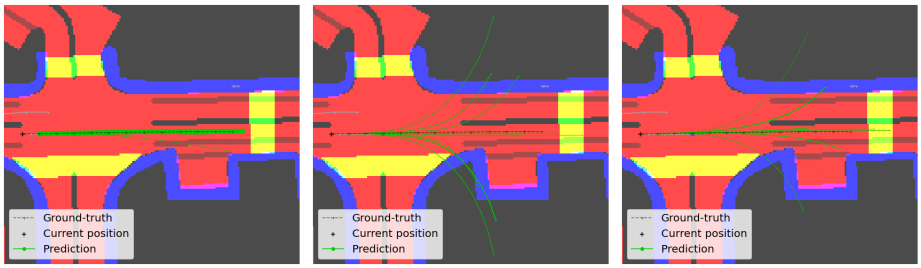
Learning-based trajectory prediction models have encountered great success, with the promise of leveraging contextual information in addition to motion history. Yet, we find that state-of-the-art forecasting methods tend to overly rely on the agent's current dynamics, failing to exploit the semantic contextual cues provided at its input. To alleviate this issue, we introduce CAB, a motion forecasting model equipped with a training procedure designed to promote the use of semantic contextual information. We also introduce two novel metrics — dispersion and convergence-to-range — to measure the temporal consistency of successive forecasts, which we found missing in standard metrics. Our method is evaluated on the widely adopted nuScenes Prediction benchmark as well as on a subset of the most difficult examples from this benchmark. The code is available at github.com/valeoai/CAB.

1. Introduction

Autonomous systems require an acute understanding of other agents' intention to plan and act safely, and the capacity to accurately forecast the motion of surrounding agents is paramount to achieving this [3,4,49]. Historically, physics-based approaches have been developed to achieve these forecasts [27]. Over the last years, the paradigm has shifted towards learning-based models [37,42]. These models generally operate over two sources of information: (1) scene information about the agent's surroundings, *e.g.* LiDAR point clouds [7, 8, 29, 36] or bird-eye-view rasters [8, 21, 29, 37, 42], and (2) motion cues of the agent, *e.g.* its instantaneous velocity, acceleration, and yaw rate [12, 37] or its previous trajectory [9, 22, 35, 42]. But despite being trained with diverse modalities as input, we remark that, in practice, these models tend to base their predictions on only one modality: the previous dynamics of the agent. Indeed, trajectory forecasting models obtain very similar performances when the scene information about the agent's surroundings is removed from the input (see section 4). This phenomenon stems from the very strong auto-correlations often exhibited in trajectories [6, 10]. For instance, when a vehicle is driving straight with a constant speed over the last seconds, situations in which the vehicle keeps driving straight with a constant speed are overwhelmingly represented; similarly, if a vehicle starts braking, its future path is very likely a stopping trajectory. As a consequence, models tend to converge to a local minimum consisting in forecasting motion based on correlations with the past motion cues only, failing to take advantage of the available contextual information [3, 11, 15, 25]. For example, in Figure 1, we observe that several predictions made by the Trajectron++ [42] model leave the driveable area which hints that the scene information was not correctly used by the model.

Such biased models relying too much on motion correlation and ignoring the scene information are unsatisfactory for several reasons. First, context holds crucial elements to perform good predictions when the target trajectory is not an extrapolation of the past motion. Indeed, a biased model will likely fail to forecast high-level behavior changes (*e.g.* start braking), when scene information is especially needed because of some event occurring in the surroundings (*e.g.* front vehicle starts braking). Leveraging context is thus paramount for motion anticipation, *i.e.* converging quickly and smoothly towards the ground truth ahead in time. Furthermore, a biased model has a flawed reasoning because it bases its predictions on motion signals rather than the underlying causes contained within the scene environment. For example, when applied on a vehicle that has started to decelerate, it will attribute its forecast on the past trajectory (*e.g.* 'The car will stop because it has started braking.') instead of the underlying reason (*e.g.* 'The car will stop because it is approaching an intersection with heavy traffic.') [33, 47]. As a direct consequence, explainability methods analyzing a biased model can lead to less satisfactory justifications. Overall, it is thus paramount for motion forecasting algorithms to efficiently leverage the contextual information and to ground motion forecasts on it.

[*] equal contribution



(a) CAB (ours) (b) Trajectron++ (c) Trajectron++ (no-context)

Figure 1. Predictions from a) CAB (ours), b) Trajectron++, and c) Trajectron++ without the context input. The thickness of trajectories represent their likelihood. Trajectron++ and its blind variant have very similar predictions and they both forecast trajectories that leave the driveable area. CAB is more consistent with the map. Sidewalks are in blue, crosswalks in yellow and driveable areas in red.

In this paper, we propose to equip a motion forecasting model with a novel learning mechanism that encourages predictions to rely more on the scene information, *i.e.* a bird-eye-view map of the surroundings and the relationships with neighboring agents. Specifically, we introduce *blind* predictions, *i.e.* predictions obtained with past motions of the agent only, without any contextual information. In contrast, the main model has access to both these inputs but is encouraged to produce motion forecasts that are different from the *blind* predictions, thus promoting the use of contextual information. Our model is called 'CAB' as it raises Context Awareness by leveraging Blind predictions. It is built on the Conditional Variational AutoEncoder (CVAE) framework, widely used in motion forecasting; in practice, it is instantiated with the Trajectron++ [42] trajectory forecasting backbone. Specifically, CAB acts on the probabilistic latent representation of the CVAE and encourages the latent distribution for the motion forecasts to be different to the latent distribution for the blind predictions. Additionally, we introduce Reweight, and RUBiZ, two alternative de-biasing strategies that are not specific to probabilistic forecasting models as they rely on loss and gradients reweighting respectively.

In motion forecast algorithms deployed in real robotic systems, when successive forecasts are done, it is desirable to have both a fast convergence towards the ground-truth, as well as a high consistency of consecutive predictions. Accordingly, we introduce two novel metrics: *convergence-to-range* and *dispersion*. These metrics aim at providing more refined measurements of how early models are able to anticipate their trajectory and how stable through time their successive predictions are.

Overall, our main contributions are as follows.

1. We target the problem of incorporating contextual information into motion forecasting architectures, as we find that state-of-the-art models overly rely on motion.

2. We present CAB, an end-to-end learning scheme that leverages blind predictions to promote the use of context.

3. In addition to standard evaluation practices, we propose two novel metrics, namely *dispersion* and *convergence-to-range*, that respectively measure the temporal stability of successive predictions and their spatial convergence speed.

To validate the design of our approach, we conduct experiments on nuScenes [6], a public self-driving car dataset focused on urban driving. We show that we outperform previous works [42, 52], as well as the alternative debiasing strategies that we propose, inspired by the recent literature in Visual Question Answering (VQA) and Natural Language Inference (NLI) [5, 31]. Besides, we use Shapley values to measure the contribution of each modality on the predictions: this allows us to measure how well a model can leverage the context input. Lastly, we conduct evaluations on a subset of the most difficult examples of nuScenes where we find that our approach is better suited to anticipate high-level behavior changes.

2. Related Work

Motion forecasting models aim to predict the future trajectories of road agents. This prediction is achieved using information from their current *motion* such as velocity, acceleration or previous trajectory, and some *contextual elements* about the scene. This context can take various forms, ranging from raw LiDAR point clouds [7, 8, 26, 29, 36, 39, 40] and RGB camera stream [26, 30, 41, 45] to more semantic representations including High-Definition maps [3,8,16,17,21,29,37,42,43,51], or detections of other agents and their motion information [12, 35, 37, 42]. Recent trajectory prediction models are designed to produce multiple forecasts, attempting to capture the multiplicity of possible futures [12, 24, 46]. Various learning setups are explored to train these models: regression in the trajectory space [9,12,14,20], spatio-temporal occupancy map prediction [3, 16, 17, 43, 49], or probabilistic methods with either implicit modelling using Generative Adversarial Networks (GANs) [18, 19, 41, 51], or explicit modelling with Conditional Variational Auto-Encoder (CVAE) [21,26,40,42,44]. Our work is based on this CVAE family of methods, which not only has provided strong results in motion forecasting but also structurally defines a separation between *high-level decision* and *low-level execution* of this decision [9].

The difficulty of efficiently leveraging contextual in-

formation in deep forecasting methods is verified in motion *planning* models that suffer from 'causal confusion' on the state variable leading to catastrophic motion drift [11, 13, 15, 25]. Moreover, models' proclivity to make reasoning shortcuts and to overlook an informative input modality is also encountered in other fields that deal with inputs of different natures, such as medical image processing [48], Visual Question Answering (VQA), or Natural Language Inference (NLI). In VQA, for instance, researchers report that models tend to be strongly biased towards the linguistic inputs and mostly ignore the visual input [1, 2, 5, 34, 38]. For example, the answer to the question "What color is the banana in the image" will be "Yellow" 90% of the time, and models will ignore the image. To alleviate this issue, some recent works propose to explicitly capture linguistic biases within a question-only branch and attempt to reduce the impact of these linguistic biases in the general model, for example through adversarial regularization [38], or with a gradient reweighting strategy during training [5]. We make a parallel between the *current motion* for trajectory forecasting and the linguistic input in VQA. Also, drawing inspiration from recent de-biasing strategies used in VQA [5, 31], we propose novel methods for motion forecasting. To the best of our knowledge, biases and statistical shortcut on the agent's dynamics have not yet been studied in the context of learning-based motion forecasting.

3. Model

The goal is to predict a distribution over possible future trajectories $\mathbf{y} = [y_1, \dots, y_T]$ of a moving agent of interest in a scene, where $y_t \in \mathbb{R}^2$ is the position of the agent t steps in the future in a bird-eye view, and T is the prediction horizon. To do so, we consider a sequence of sensor measurements $\mathcal{X}$ containing motion information (*e.g.* position, velocity, acceleration) over the H previous steps. Besides, the context $\mathcal{C}$ provides information about the static (*e.g.* driveable area, crosswalks, etc.) and dynamic (*e.g.* other agents' motion) surroundings of the agent. In this framework, a prediction model provides an estimate of $p(\mathbf{y}|\mathcal{X}, \mathcal{C})$ for any given input pair $(\mathcal{X}, \mathcal{C})$.

3.1. Conditional VAE framework for motion forecasting

Following recent works in trajectory forecasting [21, 26, 40, 42, 46, 52], we use the CVAE framework to train our model for future motion prediction. A CVAE provides an estimate $p_\Theta(\mathbf{y}|\mathcal{X}, \mathcal{C})$ of the distribution of possible trajectories by introducing a latent variable $z \in \mathcal{Z}$ that accounts for the possible high-level decisions taken by the agent:

$$p_\Theta(\mathbf{y}|\mathcal{X}, \mathcal{C}) = \int_{z \in \mathcal{Z}} p_\theta\left(z|\mathcal{X}, \mathcal{C}\right) p_\phi(\mathbf{y}|\mathcal{X}, \mathcal{C}, z), \quad (1)$$

where $\Theta = \{\theta, \phi\}$.

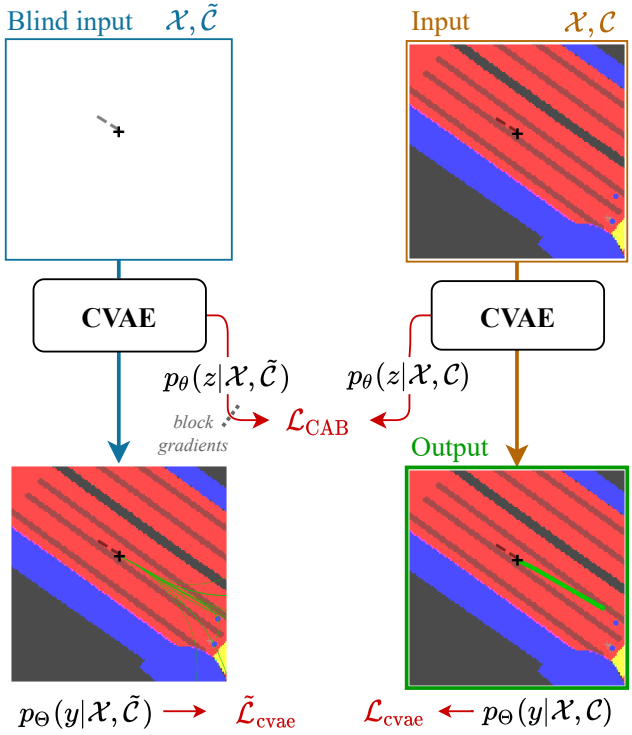


Figure 2. **Overview of the learning scheme of CAB.** CAB employs a CVAE backbone which produces distributions $p_\theta\left(z|\mathcal{X}, \mathcal{C}\right)$ and $p_\Theta(\mathbf{y}|\mathcal{X}, \mathcal{C})$ over the latent variable z and the future trajectory. During training, a *blind* input $\mathcal{X}, \tilde{\mathcal{C}}$ is forwarded into the CVAE and the resulting distribution over z is used to encourage the prediction of the model to be different from the context-agnostic distribution $p(y|\mathcal{X})$, thanks to the $\mathcal{L}_{\text{CAB-KL}}$ loss. Note that the two depicted CVAEs are identical. The original context $\mathcal{C}$ is overlayed onto the prediction for visualization purposes.

To train the CVAE, we need to estimate the latent variable z corresponding to a given trajectory $\mathbf{y}$. To that end, we introduce the additional distribution $q_\psi(z|\mathcal{X}, \mathcal{C}, \mathbf{y})$.

Distributions $p_\theta\left(z|\mathcal{X}, \mathcal{C}\right)$, $p_\phi(\mathbf{y}|\mathcal{X}, \mathcal{C}, z)$ and $q_\psi(z|\mathcal{X}, \mathcal{C}, \mathbf{y})$ are parameterized by neural networks, where θ, ϕ and ψ are their respective weights. These networks are jointly trained to minimize:

$$\begin{aligned}\mathcal{L}_{\text{cvae}} = &\frac{1}{N}\sum_{i=1}^{N} -\log p_\Theta(\mathbf{y}_i|\mathcal{X}_i, \mathcal{C}_i) \\ &+ \alpha D_{\text{KL}}[q_\psi\left(z|\mathcal{X}_i, \mathcal{C}_i, \mathbf{y}_i\right) \| \, p_\theta\left(z|\mathcal{X}_i, \mathcal{C}_i\right)],\end{aligned} \quad (2)$$

where the summation ranges over the N training samples indexed by i, and D_{KL} is the Kullback-Leibler divergence.

3.2. CAB

Using this setup, ideally, the networks would learn to extract relevant information from both motion and context to produce the most likely distribution over possible outputs

$\mathbf{y}$. However, because of the very strong correlation between $\mathcal{X}$ and $\mathbf{y}$ in driving datasets, they tend, in practice, to learn to focus essentially on $\mathcal{X}$ and to ignore $\mathcal{C}$ when estimating $p(\mathbf{y}|\mathcal{X},\mathcal{C})$. In the worse cases, models can collapse into estimating simply $p(\mathbf{y}|\mathcal{X})$. Yet, $\mathcal{C}$ contains crucial information such as road boundaries or pedestrians. Our goal is then to encourage taking $\mathcal{C}$ into account by introducing a regularization term $\mathcal{L}_{\text{CAB}}$ to the CVAE objective:

$$\mathcal{L} = \mathcal{L}_{\text{cvae}} + \mathcal{L}_{\text{CAB}}. \tag{3}$$

The idea of $\mathcal{L}_{\text{CAB}}$ is to encourage the prediction of the model to be different from $p\left(\mathbf{y}|\mathcal{X}\right)$. However, in practice, we do not have access to this distribution. Instead, we introduce a *blind-mode* for the CVAE model by simply replacing the context input $\mathcal{C}$ by a *null* context $\tilde{\mathcal{C}}$. We obtain $p_\Theta(\mathbf{y}|\mathcal{X},\tilde{\mathcal{C}})$, an explicitly flawed model whose predictions can then be used to steer the learning of the main model $p_\Theta(\mathbf{y}|\mathcal{X},\mathcal{C})$ away from focusing exclusively on $\mathcal{X}$.

To do so, we would want $\mathcal{L}_{\text{CAB}}$ to increase $D_{\text{KL}}[p_\Theta(\mathbf{y}|\mathcal{X},\mathcal{C})\|\, p_\Theta(\mathbf{y}|\mathcal{X},\tilde{\mathcal{C}})]$. Unfortunately, this term is intractable in the general case, and computing a robust Monte-Carlo estimate requires sampling a very large number of trajectories, which would significantly slow down the training. Therefore, we simplify the problem by setting this divergence constraint on the distributions over z instead of the distributions over $\mathbf{y}$. We thus minimize

$$\mathcal{L}_{\text{CAB-KL}} = -D_{\text{KL}}[p_\theta\left(z|\mathcal{X},\mathcal{C}\right)\|\, p_\theta(z|\mathcal{X},\tilde{\mathcal{C}})] \tag{4}$$

instead. Following the intuition proposed in [9], the distributions over z model intent uncertainties, whereas distributions over $\mathbf{y}$ merge intent and control uncertainties. In this case, forcing $p_\theta(z|\mathcal{X},\mathcal{C})$ and $p_\theta(z|\mathcal{X},\tilde{\mathcal{C}})$ to have a high D_{KL} explicitly sets this constraint on high-level decisions.

Moreover, to make sure that $p_\Theta(\mathbf{y}|\mathcal{X},\tilde{\mathcal{C}})$ is a reasonable approximation for $p(\mathbf{y}|\mathcal{X})$, we also optimize parameters Θ for an additional term $\tilde{\mathcal{L}}_{\text{cvae}}$, which consists in the loss described in Equation 2 where each $\mathcal{C}_i$ is replaced by $\tilde{\mathcal{C}}$.

The final $\mathcal{L}_{\text{CAB}}$ objective is then

$$\mathcal{L}_{\text{CAB}} = \lambda_{\text{KL}}\mathcal{L}_{\text{CAB-KL}} + \lambda\tilde{\mathcal{L}}_{\text{cvae}}, \tag{5}$$

where λ and λ_{KL} are hyper-parameters.

To ensure that the blind distribution focuses solely on approximating $p(\mathbf{y}|\mathcal{X})$, $\mathcal{L}_{\text{CAB-KL}}$ is only back-propagated along $p_\theta(z|\mathcal{X},\mathcal{C})$ and not along $p_\theta(z|\mathcal{X},\tilde{\mathcal{C}})$. We underline that $\mathcal{L}_{\text{CAB}}$ does not introduce extra parameters.

3.3. Instanciation of CAB with Trajectron++

To show the efficiency of CAB, we use Trajectron++ [42], a popular model for trajectory prediction based on a variant of a CVAE and whose code is freely available. We first discuss how the loss of Trajectron++ deviates from standard CVAE, and then present its implementation.

Information Maximizing Categorical CVAE Trajectron++ deviates from the standard CVAE setup in two notable ways. Firstly, following [50], they include in the CVAE objective $\mathcal{L}_{\text{cvae}}$ a mutual information term $I_q(\mathcal{X},\mathcal{C},z)$ between the inputs $(\mathcal{X},\mathcal{C})$ and the latent factor z. Secondly, in Trajectron++, the latent variable z is set as categorical. The output distribution defined in Equation 1 is then modeled as a Gaussian mixture with $|\mathcal{Z}|$ modes. These deviations are easily integrated to CAB by adding the same mutual information term and also setting z as categorical. As with Gaussian distributions that are often used in the context of VAEs, the D_{KL} between two categorical distributions has a differentiable closed-form expression.

Data and implementation in Trajectron++ The dynamic history of the agent is a multi-dimensional temporal signal $\mathcal{X} = [\mathbf{x}_{-H},...,\mathbf{x}_{-1},\mathbf{x}_0]$, where each vector $\mathbf{x}_j \in \mathbb{R}^8$ contains position, velocities, acceleration, heading and angular velocity. This sequence is encoded into a vector $\mathbf{x} = f_{\mathbf{x}}\left(\mathcal{X}\right)$, where $f_{\mathbf{x}}$ is designed as a recurrent neural network. The visual context $\mathcal{C}$ is represented by two quantities that provide external information about the scene. The first is a bird-eye view image $\mathcal{M} \in \{0,1\}^{h\times w\times l}$, constructed from a high-definition map, where each element $\mathcal{M}[h,w,l]$ encodes the presence or the absence of the semantic class l at the position (h,w). Classes correspond to semantic types such as "driveable area", "pedestrian crossing" or "walkway". This tensor $\mathcal{M}$ is processed by a convolutional neural network to provide $\mathbf{m} = f_{\mathbf{m}}(\mathcal{M}) \in \mathbb{R}^{d_m}$. The second quantity is a vector $\mathbf{g} \in \mathbb{R}^{d_g}$ that encodes the dynamic state of neighboring agents. We define the context vector $\mathbf{c}$ as the concatenation of $\mathbf{m}$ and $\mathbf{g}$.

As discussed here-above, distributions $p_\theta\left(z|\mathcal{X}_i,\mathcal{C}_i\right)$ and $q_\psi\left(z|\mathcal{X}_i,\mathcal{C}_i,\mathbf{y}_i\right)$ from Equation 2 are set as categorical distributions, parameterized by the outputs of neural networks $f_\theta(\mathbf{x},\mathbf{c})$ and $f_\psi(\mathbf{x},\mathbf{c},\mathbf{y})$ respectively.

Then, for each $z \in \mathcal{Z}$, we have $p_\phi(\mathbf{y}|\mathbf{x},\mathbf{c},z) = \mathcal{N}\left(\mu_z,\Sigma_z\right)$, where $(\mu_z,\Sigma_z) = f_\phi\left(\mathbf{x},\mathbf{c},z\right)$. These Gaussian densities are weighted by the probabilities of the corresponding z, and summed to provide the trajectory distribution:

$$p_\Theta(\mathbf{y}|\mathbf{x},\mathbf{c}) = \sum_{z\in\mathcal{Z}} p_\theta(z|\mathbf{x},\mathbf{c})p_\phi(\mathbf{y}|\mathbf{x},\mathbf{c},z). \tag{6}$$

Interestingly, f_ϕ is constructed as a composition of two functions. The first is a neural network whose output is a distribution over control values for each prediction timestep. The second is a dynamic integration module that models temporal coherence by transforming these control distributions into 2-D position distributions. This design ensures that output trajectories are dynamically feasible. For more details, please refer to [42].

3.4. Alternative de-biasing strategies

We also propose two alternative de-biaising strategies. Like CAB, they leverage blind predictions to encourage the model to use the context. However, unlike CAB that plays on the specificity of motion forecasting by acting on distribution of the latent representation, these variations are inspired by recent models from the VQA and NLI fields.

• **Reweight** is inspired by the de-biased focal loss proposed in [31]. The importance of training examples is dynamically modulated during the learning phase to focus more on examples poorly handled by the *blind* model. Formally, the model optimizes the following objective:

$$\begin{aligned}\mathcal{L}_{\text{rw}} =& \mathcal{L}_{\text{cvae}} + \tilde{\mathcal{L}}_{\text{cvae}} \\ & -\sum_{i=1}^{N} \sigma(-\log p_\Theta(\mathbf{y}_i|\mathcal{X}_i, \tilde{\mathcal{C}})) \log p_\Theta(\mathbf{y}_i|\mathcal{X}_i, \mathcal{C}_i),\end{aligned} \tag{7}$$

where σ represents the sigmoid function. Intuitively, samples that can be well predicted from the blind model, *i.e.* low value of $\sigma(-\log p_\Theta(\mathbf{y}_i|\mathcal{X}_i, \tilde{\mathcal{C}}_i))$, will see their contribution lowered and reciprocally, the ones that require contextual information to make accurate forecast, *i.e.* high value of $\sigma(-\log p_\Theta(\mathbf{y}_i|\mathcal{X}_i, \tilde{\mathcal{C}}_i))$, have an increased weight. Similarly to CAB, we prevent the gradients to flow back into the blind branch from the loss weight term.

• **RUBiZ** adjusts gradients instead of sample importance. It does so by modulating the predictions of the main model during training to resemble more to predictions of a blind model. RUBiZ is inspired by RUBi [5], a VQA model designed to mitigate language bias. Originally designed for the classification setup, we adapt this de-biasing strategy to operate over the latent factor z of our model, hence the name RUBiZ. In practice, given $\mathbf{l}$ and $\tilde{\mathbf{l}}$ the logits of $p_\theta(z|\mathcal{X}, \mathcal{C})$ and $p_\theta(z|\mathcal{X}, \tilde{\mathcal{C}})$, a new distribution over the latent variable is obtained as $p_\theta^{\text{rubiz}}(z|\mathcal{X}, \mathcal{C}, \tilde{\mathcal{C}}) = \text{softmax}(\sigma(\mathbf{l}) * \sigma(\tilde{\mathbf{l}}))$. This distribution, when used by the decoder, shifts the output of the main prediction towards a blind prediction. Consequently, situations where scene information is essential and past trajectory is not enough have increased gradient, whereas easy examples that are well predicted by the blind model have less importance in the global objective.

4. Experiments

4.1. nuScenes Dataset

Our models are trained and evaluated on the driving dataset nuScenes [6]. It contains a thousand 20-second urban scenes recorded in Boston and Singapore. Each scene includes data from several cameras, lidars, and radars, a high-definition map of the scene, as well as annotations for surrounding agents provided at 2 Hz. These annotations are processed to build a trajectory prediction dataset for surrounding agents, and especially for vehicles. Models are trained and evaluated on the official train/val/test splits from the nuScenes Prediction challenge, respectively containing 32186 / 8560 / 9041 instances, each corresponding to a specific agent at a certain time step for which we are given a 2-second history ($H = 4$) and are expected to predict up to 6 seconds in the future ($T = 12$).

4.2. Baselines and details

Physics-based baselines We consider four simple physics-based models, and a *Physics oracle*, as introduced in [37], that are purely based on motion cues and ignore contextual elements. The four physics-based models use the current velocity, acceleration, and yaw rate and forecast assuming constant speed/acceleration and yaw/yaw rate. The trajectory predicted by the *Physics oracle* model is constructed by selecting the best trajectory, in terms of average point-wise Euclidean distance, from the pool of trajectories predicted by the four aforementioned physics-based models. This *Physics Oracle* serves as a coarse upper bound on the best achievable results from a blind model that would be purely based on motion dynamics and ignores the scene structure.

Learning-based forecasting methods We compare our debiased models against recently published motion prediction models. **CoverNet** [37] forwards a rasterized representation of the scene and the vehicle state (velocity, acceleration, yaw rate) into a CNN and learns to predict the future motion as a class, which corresponds to a pre-defined trajectory. We re-train the "fixed $\epsilon = 2$" variant, for which the code is available, to compare it with our models. **Trajectron++** [42] is our baseline, which corresponds to removing $\mathcal{L}_{CAB}$ in CAB. **HalentNet** [52] casts the Trajectron++ model as the generator of a Generative Adversarial Network (GAN) [18]. A discriminator is trained to distinguish real trajectories from generated ones and to recognize which z was chosen to sample a trajectory. It also introduces 'hallucinated' predictions in the training, which correspond to predictions with several confounded values of z. To measure the usefulness of the contextual elements in these models, we also consider the 'Trajectron++ (no-context)' and 'HalentNet (no-context)' variants that simply discard the map and social interactions from the input of the respective underlying models. Trajectron++ and HalentNet are not evaluated for different temporal horizons on the nuScenes prediction challenge splits and we thus re-train them given their respective codebases.

Implementation details We use the ADAM optimizer [23], with a learning rate of 0.0003. The value of hyperparameters $\lambda = 1.0$ and $\lambda_{\text{KL}} = 5.0$ are found on the validation set.

Model	ADE-ML @1s	@2s	@3s	@4s	@5s	@6s	FDE-ML @1s	@2s	@3s	@4s	@5s	@6s	OffR-ML @6s	ADE-f @6s	FDE-f @6s	OffR-f @6s
Constant vel. and yaw	0.46	0.94	1.61	2.44	3.45	4.61	0.64	1.74	3.37	5.53	8.16	11.21	0.14	-	-	-
Physics Oracle	0.43	0.82	1.33	1.98	2.76	3.70	0.59	1.45	2.69	4.35	6.47	9.09	0.12	-	-	-
Covernet, fixed $\epsilon = 2$	0.81	1.41	2.11	2.93	3.88	4.93	1.07	2.35	3.92	5.90	8.30	10.84	**0.11**	-	-	-
Trajectron++ (no-context)	0.13	0.39	0.87	1.59	2.56	3.80	0.15	0.86	2.23	4.32	7.22	10.94	0.27	4.46	12.32	0.36
Trajectron++	0.13	0.39	0.86	1.55	2.47	3.65	0.15	0.87	2.16	4.15	6.92	10.45	0.23	4.15	11.44	0.29
HalentNet (no-context)	**0.12**	0.38	0.82	1.43	2.21	3.17	**0.13**	0.85	2.04	3.72	5.92	8.64	0.27	4.13	10.95	0.29
HalentNet	0.14	0.41	0.87	1.51	2.32	3.29	0.17	0.88	2.14	3.91	6.15	8.83	0.28	3.98	10.61	0.25
Reweight	0.13	0.38	0.81	1.42	2.20	3.14	0.15	0.83	2.00	3.69	5.90	8.58	0.17	3.71	9.74	0.19
RUBiZ	0.18	0.42	0.82	1.40	2.14	3.04	0.23	0.84	1.95	3.55	5.65	8.21	**0.11**	3.68	9.45	**0.17**
CAB	**0.12**	**0.34**	**0.73**	**1.29**	**2.01**	**2.90**	0.14	**0.73**	**1.81**	**3.39**	**5.47**	**8.02**	0.13	**3.41**	**9.03**	0.20

Table 1. **Trajectory forecasting on the nuScenes Prediction challenge [6].** Reported metrics are the Average/Final Displacement Error (ADE/FDE), and the Off-road Rate (OffR). Each metric is computed for both the most-likely trajectory (*-ML*) and the full distribution (*-f*).

4.3. Results and standard evaluations

We compare our debiased models to the baselines by measuring the widely used metrics of displacement and off-road rate. All models are trained to predict 6 seconds in the future, and their performance is evaluated for varying temporal horizons ($T \in \{2, 4, 6, 8, 10, 12\}$). Average Displacement Error (ADE) and Final Displacement Error (FDE) measure the distance between the predicted and the ground-truth trajectory, either as an average between each corresponding pair of points (ADE), or as the distance between final points (FDE). To compute these metrics with CAB, we sample the most likely trajectory $\mathbf{y}_{\text{ML}}$ by first selecting the most likely latent factor $z_{\text{ML}} = \arg\max_{z \in \mathcal{Z}} p_\theta(z|\mathbf{x}, \mathbf{c})$, and then computing the mode of the corresponding Gaussian $\mathbf{y}_{\text{ML}} = \arg\max_y p_\phi(\mathbf{y}|\mathbf{x}, \mathbf{c}, z_{\text{ML}})$. To evaluate the quality of the whole distribution and not just the most-likely trajectory, similarly to [42, 52], we compute metrics 'ADE-f' and 'FDE-f'. They are respectively the average and final displacement error averaged for 2000 trajectories randomly sampled in the full distribution predicted by the network $\mathbf{y}_{\text{full}} \sim p_\Theta(\mathbf{y}|\mathbf{x}, \mathbf{c})$. Finally, the 'off-road rate' (OffR) is the rate of future trajectories that leave the driveable area.

In Table 1, we compare the performance of our models CAB, Reweight and RUBiZ with baselines from the recent literature. To begin with, we remark that for the Trajectron++ model, the use of context brings close to no improvement for predictions up to 4 seconds and a very small one for 5- and 6-second horizons. Even more surprisingly, the HalentNet (no-context) model which does *not* use any information from the surroundings, shows better ADE-ML and FDE-ML than the regular context-aware HalentNet model. This supports our claim that the contextual elements are overlooked by these models and that predictions are mostly done by relying on motion cues. Moreover, we emphasize that the *Physics oracle* — which is purely based on motion dynamics — obtains very strong performances (3.70 ADE-ML@6s, 9.09 FDE-ML@6s) as it can choose the closest trajectory to the ground truth from a variety of dynamics. Its scores approximate upper bounds on the forecasting abilities of purely motion-based models and we observe that learning-based methods hardly outperform this *Physics-oracle* on long temporal horizons.

On the other hand, we remark that all three of our debiasing strategies significantly outperform the *Physics oracle* and previous models on almost all the metrics, both when looking at the most-likely trajectory as well as the full future distribution. This validates the idea, shared in our methods, to enforce the model's prediction to have a high divergence with a blind prediction. Indeed, despite optimizing very different objective functions, our *Reweight* and *RUBiZ* and CAB share the idea of a motion-only encoding. More precisely, at a 6-second horizon, the sample reweighting strategy gives a relative improvement of 16% w.r.t. Trajectron++. The more refined RUBiZ strategy of gradient reweighting gives a relative improvement of 19% w.r.t. Trajectron++. CAB achieves a 22% relative improvement over Trajectron++. This indicates that guiding the model's latent variable constitutes a better use of *blind* predictions than simple example or gradient weightings.

4.4. Further analyses: stability, convergence, Shapley values

We hypothesize that properly leveraging contextual information has a strong impact on the ability to anticipate the agent's intents. Intuitively, for an agent arriving at an intersection, a model without context will begin predicting a stopping trajectory only from the moment when this agent starts to stop, whereas a model with a proper understanding of contextual information will be able to foresee this behavior change ahead in time. Furthermore, improving this anticipation ability should also help the temporal stability of the predictions, as unanticipated changes of trajectory will be less likely. Unfortunately, ADE and FDE metrics do not explicitly measure the rate of convergence towards the ground-truth, nor the stability of successive predictions.

Consequently, we introduce two new metrics focusing on the stability and convergence rate of successive forecasts. Instead of classically looking at the T-step forecast

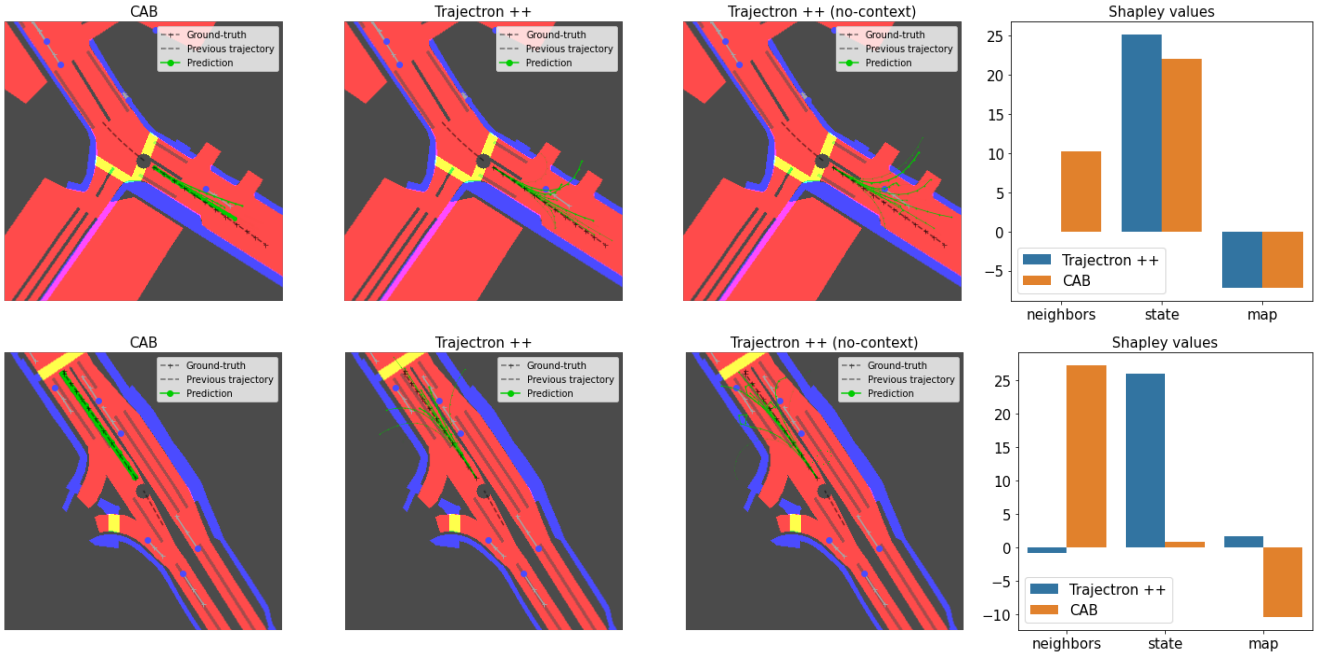


Figure 3. **Visualizations of predicted trajectories and Shapley values**. The thickness of lines represent the probability of each trajectory.

Model	Dispersion $D \downarrow$	Convergence-to-range $C(\tau) \uparrow$		
		$\tau = 20$cm	$\tau = 1$m	$\tau = 5$m
Cst. accel., yaw	6.55	0.44	1.49	3.30
Cst. accel., yaw rate	6.03	0.47	1.58	3.46
Cst. speed, yaw rate	3.42	0.54	1.82	4.15
Cst. vel., yaw	3.33	0.53	1.82	4.16
Physics Oracle	2.99	0.55	1.91	4.38
Covernet, fixed $\epsilon = 2$	4.06	0.15	0.93	3.50
Trajectron++ (no-context)	3.65	1.00	2.11	4.17
Trajectron++	3.55	0.98	2.11	4.22
Halentnet (no-context)	2.85	1.03	2.28	4.59
Halentnet	3.23	0.90	2.07	4.26
Reweight	3.02	1.03	2.27	4.50
RUBiZ	2.73	0.94	2.31	4.60
CAB	**2.61**	**1.12**	**2.45**	**4.74**

Table 2. **Study of the temporal stability of trajectory prediction**, with the Dispersion (D) and Convergence-to-range ($C(\tau)$) metrics. Predictions are made at a 6-second horizon.

$\mathbf{y}^t = [y_1^t, \dots, y_T^t]$ made at a specific time t, we take a dual viewpoint by considering the consecutive predictions $[y_T^{\hat{t}-T}, \dots, y_1^{\hat{t}-1}]$ made for the same ground-truth point $y_{\hat{t}}^{\text{gt}}$. When the agent approaches the timestamp $\hat{t}$, as t grows, predictions $y_t^{\hat{t}-t}$ will get closer to the ground-truth $y_{\hat{t}}^{\text{gt}}$. In addition to low ADE/FDE scores, it is desirable to have both (1) a high consistency of consecutive predictions, as well as (2) a fast convergence towards the ground-truth $y_{\hat{t}}^{\text{gt}}$. Therefore, for a given annotated point at $\hat{t}$, we define the *dispersion* $D_{\hat{t}}$ as the standard deviation of the points predicted by the model for this specific ground-truth point $D_{\hat{t}} = \texttt{STD}\left(\|y_t^{\hat{t}-t} - \bar{y}_{\hat{t}}\|\right)_{t\in[\![1,T]\!]}$ where $\bar{y}_{\hat{t}}$ is the barycenter of $\{y_t^{\hat{t}-t}\}_{t\in[\![1,T]\!]}$. The global dispersion score D is obtained by averaging these values over all the points in the dataset. Moreover, we propose the *convergence*-to-range-τ metric $C(\tau)_{\hat{t}}$ as the time from which all subsequent predictions fall within a margin τ of the ground-truth $y_{\hat{t}}^{\text{gt}}$, where τ is a user-defined threshold:

$$C_{\hat{t}}(\tau) = \max\left\{T' \in [\![1,T]\!] \,|\, \forall t \leq T', \|y_t^{\hat{t}-t} - y_{\hat{t}}^{gt}\|_2 \leq \tau\right\}. \tag{8}$$

In Table 2, we report evaluations of the stability and spatial convergence metrics. First, we observe that previous learning-based forecasting models have more limited anticipation capacities than the simpler physics-based models, in terms of both convergence speed (metric $C(\tau)$) and convergence stability (metric D). Consistently to the results of Table 1, we remark that our de-biased strategies, and especially our main model CAB, lead to better anticipation scores as they converge faster towards the ground truth.

In Figure 3, we visualize trajectories generated by CAB and the baselines Trajectron++ and Trajectron++ (no-context). We also analyze the contribution brought by each input of the model. To do so, we estimate the Shapley values [28] which correspond to the signed contribution of individual input features on a scalar output, the distance to the final predicted point in our case. We remark that the Shapley value of the state signal is overwhelmingly higher than the ones attributed to the map and the neighbors for Trajectron++. This means that the decisions are largely made from the agent's dynamics. This can further be seen as sev-

Model	1%	2%	3%	All
Trajectron++ (no-context)	15.80	15.58	14.97	10.94
Trajectron ++	13.12	12.69	12.25	10.45
HalentNet	14.12	12.83	12.09	8.83
Reweight	14.00	13.30	12.58	8.58
RUBiZ	13.42	12.49	11.64	8.21
CAB	**12.13**	**11.88**	**11.59**	**8.02**

Table 3. **Final Displacement Error FDE@6s on challenging situations,** as defined by Makansi et al. [32]. Results for columns '$i\%$' are averaged over the top $i\%$ hardest situations as measured by the the mismatch between the prediction from a Kalman filter and the ground-truth.

eral predicted trajectories are highly unlikely futures as they collide with the other agents and/or leave the driveable area. Instead, the Shapley values for CAB give much more importance to both the map and the neighboring agents, which helps to generate likely and acceptable futures.

4.5. Evaluation on hard situations

We verify that the performance boost observed in Table 1 does not come at the expense of a performance drop on difficult yet critical situations. Accordingly, we use recently proposed evaluations [32] as they remark that uncritical cases dominate the prediction and that complex scenarios cases are at the long tail of the dataset distribution. In practice, situations are ranked based on how well the forecast made by a Kalman filter fits the ground-truth trajectory.

In Table 3, we report such stratified evaluations, on the 1%, 2%, and 3% hardest situations. Our first observation is that the overall performance ('All') does not necessarily correlate with the capacity to anticipate hard situations. Indeed, while HalentNet significantly outperfoms Trajectron++ on average, it falls short on the most challenging cases. Besides, CAB achieves better results than Trajectron++ on the hardest situations (top 1%, 2%, and 3%). Lastly, while the gap between Trajectron++ and CAB is only 0.66 point for the 3% of hard examples, it increases for the top 1% of hardest examples up to 0.99.

In Figure 4, we display some qualitative results we obtain on challenging situations selected among the 1% hardest examples. On the left, we observe that the turn is not correctly predicted by Trajectron++ as it estimates several possible futures that leave the driveable area. On the right, the agent of interest has to stop because of stopped agents in front of it and this behavior is well forecasted by CAB, unlike Trajectron++ which extrapolates the past and provides multiple futures colliding into other agents. Overall, the better use of the context in CAB not only helps on average situations but also on difficult and potentially critical ones.

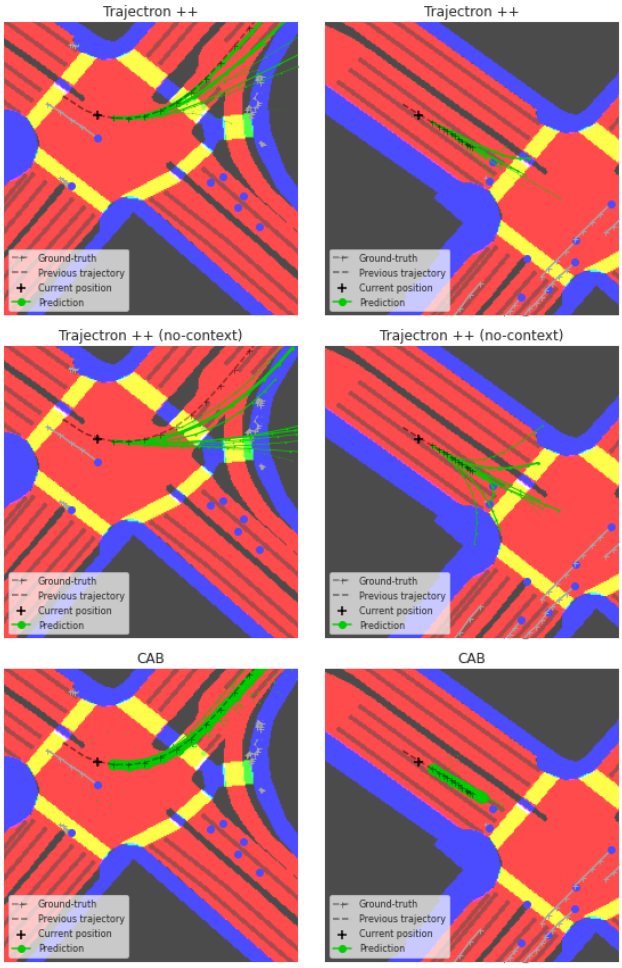


Figure 4. **Visualizations on challenging situations,** as defined by Makansi et al. [32]. By better leveraging the context, CAB generates more accurate predictions while Trajectron++ leaves the driveable area or collides into other agents.

5. Conclusion

We showed that modern motion forecasting models struggle to use contextual scene information. To address this, we introduced *blind* predictions that we leveraged with novel de-biaising strategies. This results into three motion forecasting models designed to focus more on context. We show that doing so helps reducing statistical biases from which learning-based approaches suffer. In particular, CAB, which is specifically built for probabilistic forecasting models, makes significant improvements in traditional distance-based metrics. Finally, after introducing new stability and convergence metrics, we show that CAB shows better anticipation properties than concurrent methods.

Acknowledgments: We thank Thibault Buhet, Auguste Lehuger and Ivan Novikov for insightful comments. This work was supported by ANR grant VISA DEEP (ANR-20-CHIA-0022) and MultiTrans (ANR-21-CE23-0032).

References

[1] Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. In *EMNLP*, 2016. 3

[2] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, 2018. 3

[3] Mayank Bansal, Alex Krizhevsky, and Abhijit S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Robotics: Science and Systems XV*, 2019. 1, 2

[4] Thibault Buhet, Émilie Wirbel, and Xavier Perrotton. PLOP: probabilistic polynomial objects trajectory planning for autonomous driving. In *CoRL*, Proceedings of Machine Learning Research, 2020. 1

[5] Rémi Cadène, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases for visual question answering. In *NeurIPS*, 2019. 2, 3, 5

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 1, 2, 5, 6

[7] Sergio Casas, Cole Gulino, Renjie Liao, and Raquel Urtasun. Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. In *ICRA*, 2020. 1, 2

[8] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *CoRL*, 2018. 1, 2

[9] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *CoRL*, Proceedings of Machine Learning Research, 2019. 1, 2, 4

[10] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *CVPR*, 2019. 1

[11] Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *ICCV*, 2019. 1, 3

[12] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *ICRA*, 2019. 1, 2

[13] Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *NeurIPS*, 2019. 3

[14] Nachiket Deo and Mohan M. Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *CVPR Workshops*, 2018. 2

[15] Laurent George, Thibault Buhet, Émilie Wirbel, Gaetan Le-Gall, and Xavier Perrotton. Imitation learning for end to end vehicle longitudinal control with forward camera. *CoRR*, abs/1812.05841, 2018. 1, 3

[16] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. GOHOME: graph-oriented heatmap output for future motion estimation. *CoRR*, abs/2109.01827, 2021. 2

[17] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. HOME: heatmap output for future motion estimation. In *IITSC*, 2021. 2

[18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 2, 5

[19] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018. 2

[20] Joey Hong, Benjamin Sapp, and James Philbin. Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In *CVPR*, 2019. 2

[21] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *ICCV*, 2019. 1, 2, 3

[22] Byeoungdo Kim, Chang Mook Kang, Jaekyum Kim, Seung-Hi Lee, Chung Choo Chung, and Jun Won Choi. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In *ITSC*, 2017. 1

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[24] Kris M. Kitani, Brian D. Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *ECCV*, 2012. 2

[25] Yann LeCun, Urs Muller, Jan Ben, Eric Cosatto, and Beat Flepp. Off-road obstacle avoidance through end-to-end learning. In *NIPS*, 2005. 1, 3

[26] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H. S. Torr, and Manmohan Chandraker. DESIRE: distant future prediction in dynamic scenes with interacting agents. In *CVPR*, 2017. 2, 3

[27] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal*, 2014. 1

[28] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NIPS*, 2017. 7

[29] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *CVPR*, 2018. 1, 2

[30] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *AAAI*, 2019. 2

[31] Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. End-to-end bias mitigation by modelling biases in corpora. In *ACL*, 2020. 2, 3, 5

[32] Osama Makansi, Özgün Çiçek, Yassine Marrakchi, and Thomas Brox. On exposing the challenging long tail in future prediction of traffic actors. In *ICCV*, 2021. 8

[33] Osama Makansi, Julius Von Kügelgen, Francesco Locatello, Peter Vincent Gehler, Dominik Janzing, Thomas Brox, and Bernhard Schölkopf. You mostly walk alone: Analyzing feature attribution in trajectory prediction. In *ICLR*, 2022. 1

[34] Varun Manjunatha, Nirat Saini, and Larry S. Davis. Explicit bias discovery in visual question answering models. In *CVPR*, 2019. 3

[35] Kaouther Messaoud, Nachiket Deo, Mohan M. Trivedi, and Fawzi Nashashibi. Multi-head attention with joint agent-map representation for trajectory prediction in autonomous driving. *CoRR*, abs/2005.02545, 2020. 1, 2

[36] Gregory P. Meyer, Jake Charland, Shreyash Pandey, Ankit Laddha, Shivam Gautam, Carlos Vallespi-Gonzalez, and Carl K. Wellington. Laserflow: Efficient and probabilistic object detection and motion forecasting. *IEEE Robotics Autom. Lett.*, 2021. 1, 2

[37] Tung Phan-Minh, Elena Corina Grigore, Freddy A. Boulton, Oscar Beijbom, and Eric M. Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *CVPR*, 2020. 1, 2, 5

[38] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. In *NeurIPS*, 2018. 3

[39] Nicholas Rhinehart, Kris M. Kitani, and Paul Vernaza. r2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *ECCV*, 2018. 2

[40] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. PRECOG: prediction conditioned on goals in visual multi-agent settings. In *ICCV*, 2019. 2, 3

[41] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. In *CVPR*, 2019. 2

[42] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, 2020. 1, 2, 3, 4, 5, 6

[43] Maximilian Schäfer, Kun Zhao, Markus Bühren, and Anton Kummert. Context-aware scene prediction network (caspnet). *CoRR*, abs/2201.06933, 2022. 2

[44] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, 2015. 2

[45] Shashank Srikanth, Junaid Ahmed Ansari, Karnik Ram R., Sarthak Sharma, J. Krishna Murthy, and K. Madhava Krishna. INFER: intermediate representations for future prediction. In *IROS*, 2019. 2

[46] Yichuan Charlie Tang and Ruslan Salakhutdinov. Multiple futures prediction. In *NeurIPS*, 2019. 2, 3

[47] Éloi Zablocki, Hedi Ben-Younes, Patrick Pérez, and Matthieu Cord. Explainability of vision-based autonomous driving systems: Review and challenges. *CoRR*, abs/2101.05307, 2021. 1

[48] John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric K. Oermann. Confounding variables can degrade generalization performance of radiological deep learning models. *CoRR*, abs/1807.00431, 2018. 3

[49] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *CVPR*, 2019. 1, 2

[50] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Balancing learning and inference in variational autoencoders. In *AAAI*, 2019. 4

[51] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris L. Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *CVPR*, 2019. 2

[52] Deyao Zhu, Mohamed Zahran, Li Erran Li, and Mohamed Elhoseiny. Halentnet: Multimodal trajectory forecasting with hallucinative intents. In *ICLR*, 2021. 2, 3, 5, 6