EMPERROR: A Flexible Generative Perception Error Model for Probing Self-Driving Planners

Niklas Hanselmann[1,2,3], Simon Doll[1,2], Marius Cordts[1], Hendrik P.A. Lensch[2] and Andreas Geiger[2,3]

***Abstract*—To handle the complexities of real-world traffic, learning planners for self-driving from data is a promising direction. While recent approaches have shown great progress, they typically assume a setting in which the ground-truth world state is available as input. However, when deployed, planning needs to be robust to the long-tail of errors incurred by a noisy perception system, which is often neglected in evaluation. To address this, previous work has proposed drawing adversarial samples from a perception error model (PEM) mimicking the noise characteristics of a target object detector. However, these methods use simple PEMs that fail to accurately capture all failure modes of detection. In this paper, we present EMPERROR, a novel transformer-based generative PEM, apply it to stress-test an imitation learning (IL)-based planner and show that it imitates modern detectors more faithfully than previous work. Furthermore, it is able to produce realistic noisy inputs that increase the planner's collision rate by up to 85 %, demonstrating its utility as a valuable tool for a more complete evaluation of self-driving planners.**

***Index Terms*—Deep Learning Methods; Object Detection, Segmentation and Categorization; Autonomous Agents**

I. INTRODUCTION

AFTER years of progress, autonomous driving systems are able to handle increasingly complex situations [1]. This is enabled, in part, by solving several aspects of driving with learned modules, such as perception [2], [3], [4] and motion forecasting [5], [6]. Recently, there has been increased interest in managing the complexity of human behavior in traffic by also learning the planning task [7], [8], which has been accelerated through the emergence of motion forecasting- and planning-centric benchmarks and datasets [9], [10]. Most of this work assumes a simplified setting where the ground-truth world state is available as input and focuses on accuracy of the planned trajectory with respect to human driving demonstrations. As a result, robustness to the residual risk of errors in the perception system, which is ultimately just an imperfect model operating on incomplete observations of the world, remains underexplored.

Manuscript received: November, 3rd, 2024; Revised March, 7th, 2025; Accepted April, 2nd, 2025.

This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments. Niklas Hanselmann and Simon Doll were supported by the German Federal Ministry for Economic Affairs and Climate Action (KI Delta Learning: project number 19A19013A). Andreas Geiger was supported by the ERC Starting Grant LEGO-3D (850533) and the DFG EXC number 2064/1 - project number 390727645.

[1] Mercedes-Benz AG R&D, Sindelfingen, Germany first.last@mercedes-benz.com

[2] University of Tübingen, Tübingen, Germany

[3] Tübingen AI Center, Tübingen, Germany

Digital Object Identifier (DOI): 10.1109/LRA.2025.3562789.

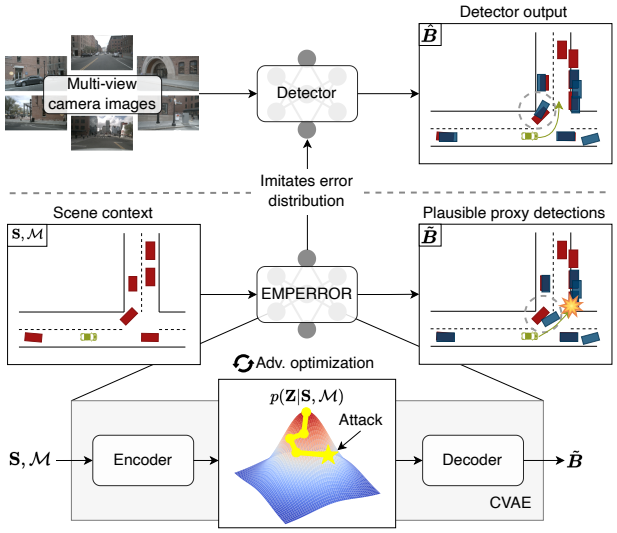


Fig. 1. **Method Overview.** We propose EMPERROR, a generative model that imitates a given detector by modeling the distribution of its perception errors conditioned on a ground-truth state and BEV map as scene context. Adversarial optimization in the model's latent space can then produce challenging yet plausible proxy detections from that distribution which stress-test the robustness of a given planner, e.g. by inducing collisions.

In this work, we aim to illuminate the susceptibility of learned planning, which is often brittle in the face of o.o.d inputs [11], to these errors. Recent seminal studies [12], [13] have approached this problem from the lens of adversarial attacks. To this end, they first construct a perception error model (PEM) [14], [15], which allows sampling multiple noisy estimates imitating a target 3D object detector given a ground-truth scene representation as context. Then, by leveraging the PEM as a proxy of the detector, challenging samples that stress-test the target planner can be found by employing an adversarial search strategy. While promising, these works consider simple, synthetic scenes that do not capture the complexities of real-world data. Moreover, they employ simple PEMs that phrase the error modeling task as an isolated, per-object perturbation of the ground-truth state, rather than jointly reasoning over the entire scene context. Hence, they cannot faithfully model the intricacies of the error patterns exhibited by modern 3D object detectors, such as duplicate detections resulting in false-positives and correlations in errors for groups of objects.

Motivated by this, we propose EMPERROR, a novel generative **emp**irical **error** model based on the transformer architecture, that can more faithfully capture the error characteristics of a target detector. Our key idea is to leverage the attention

mechanism and a flexible set of latent queries to model the full range of failure modes, including false-positives, in a scene-consistent manner. Furthermore, EMPERROR provides a prior over different error patterns for a given scene context, enabling us to draw adversarial, yet plausible samples to stress-test a target planner. Building on these advantages, we design a framework to probe the robustness of learned planners to noisy perception inputs, which is visualized in Fig. 1. We then apply the proposed framework to an imitation learning (IL)-based planner, modeling three different modern camera-based 3D object detectors, and show learned planning is indeed vulnerable to plausible noise from the long-tail of perception errors. We believe EMPERROR can serve as a valuable tool for data-driven evaluation of self-driving planners.

Contributions: (1) We propose EMPERROR, a novel flexible, transformer-based, generative PEM for probing planning, that can more faithfully imitate modern object detectors than previous work. (2) We propose and integrate EMPERROR into a framework for probing the robustness of a planner to noise in its perception system. (3) We demonstrate that this framework can reveal unsafe behavior in learned planning, even for minor, plausible detection errors.

II. RELATED WORK

Perception Error Models: Accurately modeling the noise characteristics of a perception module enables an analysis of typical failure modes and informs the design of downstream modules. Several works [16], [14], [17] rely on classical statistical models often coupled with elements of manual design to capture regression errors and false-negatives in detection tasks. While these approaches yield lightweight, easily interpretable PEMs, they are limited in the fidelity and complexity of noise patterns that can be modeled. Recent work addresses this limitation by instantiating the PEM through a neural network. In [12], the authors use a feed-forward network to model false-negative detections. In [15], [13], the authors model spatial errors and false-negatives using a probabilistic feed-forward network to enable efficient evaluation of decision making modules in simulation. In [18], [19], the authors propose fully-convolutional PEMs that mimic the error characteristics of LiDAR-based detectors [20], [21]. As they resemble the overall architecture of a standard convolutional one-stage detector, they can model regression errors, false-negatives and false-positives but are not probabilistic and thus do not permit sampling. In summary, the PEMs proposed in previous work either do not fully capture the error characteristics of a target perception system, do not permit sampling, or both. As these are prerequisites to find plausible worst-case perception errors for planning, we propose a novel PEM that satisfies these requirements.

Generating Safety-Critical Scenarios: When deployed, self-driving systems are required to robustly handle rare and potentially safety-critical scenarios from the long-tail of driving. Since real-world data collection is limited in scalability and diversity, there has been increased interest in the automated generation of safety-critical scenarios in recent years. The majority of this work focuses on automatically altering the behavior of other traffic participants to induce failure in the target autonomy system [22], [23], [24], [25], [26]. Rather than testing against external long-tail behavior, we instead look inward to examine the effects of long-tail noise in the autonomy system's own perception system by sampling from a PEM. While this high-level idea has been explored before, previous work [12], [13] uses simple PEMs that do not fully capture all dependencies and failure modes, and tests in simple, synthetic scenes that do not reflect the complexity of real-world data. In contrast, we propose a novel transformer-based PEM that fully models the false-positive, false-negative and regression error characteristics of a given 3D-object detector and apply our framework on challenging real-world data.

III. METHOD

Problem Statement: We are interested in supplementing offline evaluation by stress-testing modular autonomy systems, which we will consider to consist of a 3D object detector as the perception module and a downstream planner in this study. Let us denote the true world state as a set of vectors $\mathbf{S} = \{\mathbf{s}_0, \cdots, \mathbf{s}_N\} \in \mathbb{R}^{d_s}$ describing the position, heading angle, spatial dimensions, first- and second-order longitudinal- and angular dynamics in the ego vehicle's frame as well as the semantic class for each of N objects in the scene. The object detector then processes raw sensor observations of the world into a set of 3D bounding boxes $\hat{\mathbf{B}} = \{\hat{\mathbf{b}}_0, \cdots, \hat{\mathbf{b}}_{\hat{N}}\} \in \mathbb{R}^{d_b}$ similar to $\mathbf{S}$, but with the dynamics reduced to a velocity vector. Based on the 3D bounding boxes, as well as a rasterized bird's-eye view (BEV) map $\mathcal{M} \in \mathcal{R}^{h \times w \times 5}$ containing information on the static scene layout, the planner then computes a future trajectory $\boldsymbol{\tau}$ to be driven.

The detector incurs errors in the form of false-positives, false-negatives and inaccurate regression of bounding box parameters when compared to the ground-truth set derived directly from $\mathbf{S}$. These errors are individual to the specific model architecture and weights in question, and often depend on the configuration of the scene, e.g. due to correlations with the relative object pose, correlations among groups of objects or correlations between objects and map elements. To probe the planner's sensitivity to this imperfect perception, we would like to find worst-case, but plausible errors for a given scene that drive the planned trajectory towards violation of safety constraints. To this end, we learn a conditional generative PEM as a proxy that imitates the detector given the ground-truth scene context $(\mathbf{S}, \mathcal{M})$ as input. By repeatedly drawing samples $\tilde{\mathbf{B}} = \{\tilde{\mathbf{b}}_0, \cdots, \tilde{\mathbf{b}}_{\tilde{N}}\} \in \mathbb{R}^{d_b}$ from this proxy and measuring the quality of the corresponding planned trajectory, we aim to gauge the influence of noise patterns and optimize for planning failures, such as a collision.

A. Conditional Generative PEM

We cast the imitation of perception errors given a ground-truth scene state as a set-to-set modeling problem, for which the transformer architecture is a natural choice due to its permutation-invariant modeling of relationships among set

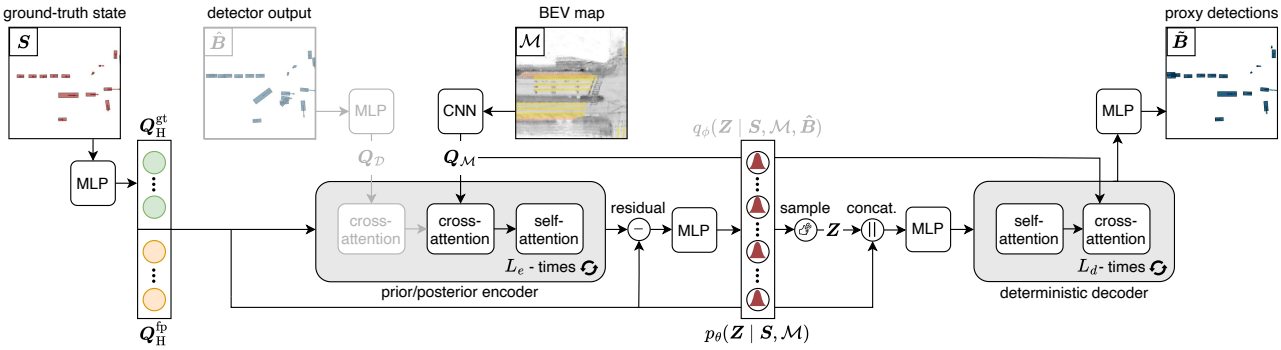


Fig. 2. **Generative Perception Error Model (PEM).** We propose a PEM based on the conditional variational autoencoder (CVAE) framework to model the error characteristics of a target detector. It consists of a *prior encoder*, inferring a distribution over the latent variable $\mathbf{Z}$ given a ground-truth state $\mathbf{S}$ and BEV map $\mathcal{M}$ as scene context at test-time, and a *deterministic decoder*, which produces a set of proxy detections $\hat{\mathbf{B}}$ given $\mathbf{Z}$. At training-time, a *posterior encoder* with a similar architecture as the prior encoder is used to encode and sample from the latent distribution. In contrast to the prior encoder, it also has access to privileged information in the form of the detector outputs $\hat{\mathbf{B}}$ to be reconstructed. Privileged components are shown in faint coloring.

elements. We propose a conditional generative model building on a transformer encoder-decoder architecture [27] under the framework of CVAEs [28]. Specifically, we aim to construct a latent variable model capturing

$$P\left(\hat{\mathbf{B}} \mid \mathbf{S}, \mathcal{M}\right) = \int P\left(\hat{\mathbf{B}} \mid \mathbf{S}, \mathcal{M}, \mathbf{Z}\right) P\left(\mathbf{Z} \mid \mathbf{S}, \mathcal{M}\right) \, d\mathbf{Z} \tag{1}$$

where $\mathbf{Z} = \{\mathbf{z}_0, \cdots, \mathbf{z}_{\hat{N}}\} \in \mathbb{R}^{d_z}$ is a set of per-object latent variables of dimensionality d_z explaining the stochasticity of the target detector's perception errors by capturing different noise patterns. This allows sampling multiple plausible sets of detections $\tilde{\mathbf{B}} \sim p_\theta(\hat{\mathbf{B}} \mid \mathbf{S}, \mathcal{M})$ by applying different latent error characteristics sampled from a learned conditional Gaussian prior modeling $\mathbf{Z} \sim p_\theta(\mathbf{Z} \mid \mathbf{S}, \mathcal{M})$ to a given scene context $(\mathbf{S}, \mathcal{M})$. This is done via a deterministic decoder $\hat{\mathbf{B}} = f_\theta(\mathbf{Z}, \mathbf{S}, \mathcal{M})$. As evaluating the integral in (1) is intractable, learning under the CVAE framework utilizes a learned approximation to the posterior $q_\phi(\mathbf{Z} \mid \mathbf{S}, \mathcal{M}, \hat{\mathbf{B}})$ to maximize the evidence lower bound (ELBO) [29], [28]:

$$\begin{aligned}\log P\left(\hat{\mathbf{B}} \mid \mathbf{S}, \mathcal{M}\right) \geq \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{S},\mathcal{M},\hat{\mathbf{B}})}\left[\log p_\theta\left(\hat{\mathbf{B}} \mid \mathbf{S}, \mathcal{M}, \mathbf{Z}\right)\right] \\ - D_{\mathrm{KL}}\left(q_\phi\left(\mathbf{Z} \mid \mathbf{S}, \mathcal{M}, \hat{\mathbf{B}}\right) \,||\, p_\theta\left(\mathbf{Z} \mid \mathbf{S}, \mathcal{M}\right)\right)\end{aligned} \tag{2}$$

Although both the prior and the posterior distribution are factorized over objects, the distribution over each latent variable $\mathbf{z}_n \in \mathbf{Z}$ is computed from the full set of ground-truth states $\mathbf{S}$ (and detector outputs $\hat{\mathbf{B}}$ for the posterior). Similarly, the decoder considers the joint sets of per-object latent variables $\mathbf{Z}$ and states $\mathbf{S}$ to output individual detections $\tilde{\mathbf{b}}_n \in \tilde{\mathbf{B}}$. This enables our model to generate realistic, scene-consistent noise patterns by allowing it to capture higher-order relationships. In the following, we briefly describe the main components in our architecture, as depicted in Fig. 2.

Initial Hypotheses: Both the probabilistic encoders modeling p_θ and q_ϕ as well as the deterministic decoder f_θ are transformers operating on a shared set of queries $\mathbf{Q}_H = \{\mathbf{q}_H^0, \cdots, \mathbf{q}_H^{N_q}\} \in \mathbb{R}^{d_h}$ representing detection hypotheses. Specifically, to model true-positives and false-negatives, we initialize a subset $\mathbf{Q}_H^{gt}$ of N_q^{gt} queries by projecting the ground-truth state of the scene to the feature dimensionality d_h of the model via a multilayer perceptron (MLP), which implies that N_q^{gt} is variable depending on the number of objects in the scene. Under the assumption that for any reasonable detector the majority of ground-truth objects will have a corresponding detector output, this provides the model with a sensible first initialization and simplifies its task to the estimation of a residual to the ground-truth regression parameters and semantic class scores. To model false-positives, a second subset $\mathbf{Q}_H^{fp}$ comprised of a fixed number of N_q^{fp} constant learnable embeddings of dimensionality d_h capturing dataset-level statistics is used as these can not be trivially initialized from the scene context.

Prior Encoder: The prior encoder refines the initial hypotheses $\mathbf{Q}_H$ while considering the context of the scene layout provided by the map $\mathcal{M}$ through a series of L_e transformer layers to estimate the parameters of the prior distribution $p_\theta(\mathbf{Z} \mid \mathbf{S}, \mathcal{M}) = \mathcal{N}(\boldsymbol{\mu}_\theta^p(\mathbf{S}, \mathcal{M}), \boldsymbol{\sigma}_\theta^p(\mathbf{S}, \mathcal{M}))$. The map is represented by a set of feature vectors $\mathbf{Q}_\mathcal{M} = \{\mathbf{q}_\mathcal{M}^0, \cdots, \mathbf{q}_\mathcal{M}^{h \times w}\} \in \mathbb{R}^{d_h}$ obtained as the cells of the feature grid of a convolutional map encoder, to which we add sinusoidal positional embeddings to retain spatial information. Through repeated blocks of self-attention among the detection hypotheses and cross-attention to the map features, the initial hypotheses are iteratively adjusted towards representing noisy detections for the current scene context. This allows the model to capture crucial relationships, such as occlusion between objects or duplicate detection hypothesis for the same ground-truth object, for example of different semantic classes or at various depths along the viewing ray, a characteristic pattern in camera-based 3D detectors. The residual between the initial and refined hypotheses, which captures possible error patterns, is then input to an MLP estimating the mean vectors and diagonal covariance matrices of the factorized prior distribution.

Posterior Encoder: The posterior encoder estimates the parameters of the approximate posterior $q_\phi(\mathbf{Z} \mid \mathbf{S}, \mathcal{M}, \hat{\mathbf{B}}) = \mathcal{N}(\boldsymbol{\mu}_\phi^q(\mathbf{S}, \mathcal{M}, \hat{\mathbf{B}}), \boldsymbol{\sigma}_\phi^q(\mathbf{S}, \mathcal{M}, \hat{\mathbf{B}}))$, which is analogous to the prior, and largely follows the same architecture. However, we

incorporate the detector outputs $\hat{\mathbf{B}}$ via an additional cross-attention block in each of its L_e transformer layers. To this end, we project them to the model's feature space similar to $\mathbf{Q}_H^{gt}$ via a separate MLP to obtain a set of detection features $\mathbf{Q}_D = \{\mathbf{q}_D^0, \cdots, \mathbf{q}_D^{N_D}\} \in \mathbb{R}^{d_h}$.

Decoder: Given the set of initial hypotheses and map context, the decoder applies a sampled latent error pattern $\mathbf{Z}$ to produce a set of proxy bounding boxes $\tilde{\mathbf{B}} = f_\theta(\mathbf{Z}, \mathbf{S}, \mathcal{M})$. To this end, the information in both $\mathbf{Q}_H$ and $\mathbf{Z}$ is first fused by applying an MLP to the feature-wise concatenation of each pair of vectors $(\mathbf{q}_n, \mathbf{z}_n)$, forming $\tilde{\mathbf{Q}}_H \in \mathcal{R}^{d_h}$. Similarly to the prior encoder, $\tilde{\mathbf{Q}}_H$ is then refined through a series of L_d transformer layers performing self-attention and cross-attending to $\mathbf{Q}_\mathcal{M}$. Finally, the refined latent queries are decoded to the bounding box regression parameters and independent per-class sigmoid classification scores. This mirrors the prevalent output configuration used in state-of-the-art 3D detectors [2], [30], [3].

Training: We optimize the model on a dataset of tuples $(\mathbf{S}, \mathcal{M}, \hat{\mathbf{B}})$ collected by running inference with a target detector on the corresponding sensor data to obtain noisy detections. Similar to prior work [31], [25], we use a modified CVAE objective $\mathcal{L}_{\text{cvae}} = \mathcal{L}_{\text{recon}} + \beta\mathcal{L}_{\text{JS}}$ consisting of a reconstruction error and a divergence regularizer:

$$\mathcal{L}_{\text{recon}} = \sum_{(i,j)\in\Omega} \left(\|\tilde{\mathbf{b}}_i^{reg} - \hat{\mathbf{b}}_j^{reg}\|_1 + \sum_{c=0}^{N_c} \text{BCE}\left(\tilde{\mathbf{b}}_{i,c}^{cls}, \hat{\mathbf{b}}_{j,c}^{cls}\right) \right) - \sum_{i\in\emptyset} \sum_{c=0}^{N_c} \log(1 - \tilde{\mathbf{b}}_{i,c}^{cls}) \tag{3}$$

$$\mathcal{L}_{\text{JS}} = \text{JS}^{G_\alpha}\left(q_\phi\left(\mathbf{Z} \mid \mathbf{S}, \mathcal{M}, \hat{\mathbf{B}}\right) \mid\mid p_\theta\left(\mathbf{Z} \mid \mathbf{S}, \mathcal{M}\right)\right) . \tag{4}$$

For the reconstruction loss, we first compute correspondences between the set of boxes $\tilde{\mathbf{B}}$ drawn from the PEM and the set of boxes $\hat{\mathbf{B}}$ produced by the target detector. The correspondences for the subset of boxes produced from ground-truth initialized hypotheses $\mathbf{Q}_H^{gt}$ are obtained by greedily matching detections $\hat{\mathbf{b}}_n$ to ground-truth bounding boxes $\mathbf{b}_n$ via the logic proposed in [32] and kept fixed throughout training. For those left unmatched, which includes the boxes produced from $\mathbf{Q}_H^{fp}$, we obtain an assignment of the remaining boxes in $\hat{\mathbf{B}}$ that have not been matched in the previous step online via the Hungarian algorithm, following the standard practice in [2], [30], [3]. This set of correspondences between predicted hypothesis and detector targets is termed Ω. Any boxes in $\tilde{\mathbf{B}}$ left without correspondence after these steps are treated as belonging to the no-object set $\emptyset$.

The reconstruction loss is then computed as the sum of an l_1-term for the regression parameters and the binary cross-entropy for the per-class sigmoid scores. Unmatched proxy bounding boxes are expected to have a classification score of zero. To discourage the prior encoder from placing probability mass in regions assigned low likelihood by the privileged approximate posterior, we replace the forward KL-divergence in (2) with the skew-geometric Jensen-Shannon divergence JS^{G_α} [33]. Intuitively, this allows us to interpolate between the forward KL, which encourages *mode-covering* behavior, and the backward KL, which encourages *mode-seeking* behavior, via a parameter α. This formulation permits trading off diversity for a decreased chance of sampling latent error patterns from the prior that would be unlikely under the posterior at test time. Finally, we use a weighting factor β to control the strength of the divergence regularizer in the overall objective, as proposed in [34].

B. Imitation Learning-based Planner

We consider a simple transformer-based planner that is prototypical of the planning modules proposed in recent literature [7], [35], [36], [37]. It operates on BEV projections of $\hat{\mathbf{B}}$ or $\tilde{\mathbf{B}}$, which are encoded via an MLP to form $\mathbf{Q}_H^\pi \in \mathbb{R}^{d_\pi}$, as well as a set of map features $\mathbf{Q}_\mathcal{M}^\pi \in \mathbb{R}^{d_\pi}$ obtained from a convolutional encoder. The planner then forms a latent plan by refining an initial constant learnable embedding $\mathbf{q}_{\text{ego}}^\pi$ through a series of L_π transformer layers consisting of two cross-attention blocks attending to the encoded detection results $\mathbf{Q}_H^\pi$ and map features $\mathbf{Q}_\mathcal{M}^\pi$. To provide additional context, we add embeddings of the current speed of the ego vehicle ϑ^{ego}, as well as a high-level navigation command c^{nav} (i.e. *go-straight*, *turn-left* or *turn-right*) to the initial $\mathbf{q}_{\text{ego}}^\pi$. Finally, an MLP decodes a trajectory $\hat{\boldsymbol{\tau}} = \pi_\omega(\mathbf{B}, \mathcal{M}, \vartheta^{\text{ego}}, c^{\text{nav}}) \in \mathbb{R}^{T_{\text{plan}}\times 3}$ of future waypoints consisting of a BEV position and heading angle. The model is trained via standard imitation-learning on expert trajectories using an l_1-loss.

C. Probing Planning

Given a scene- $(\mathbf{S}, \mathcal{M})$ and planning context $(\vartheta^{\text{ego}}, c^{\text{nav}})$, we now aim to probe π_ω with respect to its sensitivity to noise in its input perception results. To this end, we leverage our generative PEM in an adversarial fashion to draw samples $\tilde{\mathbf{B}} \sim p_\theta(\hat{\mathbf{B}} \mid \mathbf{S}, \mathcal{M})$ that induce failure in π_ω. Specifically, for a driving cost $\mathcal{C}$ measuring the quality of the generated trajectory, we formulate this process as an optimization problem:

$$\tilde{\mathbf{B}}^* = \underset{\mathbf{Z}\sim p_\theta(\mathbf{Z}|\mathbf{S},\mathcal{M})}{\text{argmax}} \; \mathcal{C}\left(\pi_\omega(f_\theta\left(\mathbf{Z}, \mathbf{S}, \mathcal{M}\right), \mathcal{M}, \vartheta^{\text{ego}}, c^{\text{nav}}), \mathbf{S}, \mathcal{M}\right) \quad \text{s.t.} \quad -\kappa\,\boldsymbol{\sigma}_{\theta,n}^p \le \mathbf{z}_n - \boldsymbol{\mu}_{\theta,n}^p \le \kappa\,\boldsymbol{\sigma}_{\theta,n}^p \quad \forall\ \mathbf{z}_n \in \mathbf{Z} \tag{5}$$

Here, the latent space of our generative model forms the search space, which is particularly amenable to adversarial optimization due to the explicit prior over $\mathbf{Z}$: By bounding the maximum standard deviation we allow each $\mathbf{z}_n$ to move from the mean, the solution space can be constrained to high-likelihood regions, ensuring plausibility.

Objective: For this study, we choose a simple collision cost measuring the closest Euclidean distance $d(\hat{\boldsymbol{\tau}}^t, \mathbf{s}_n^t)$ between the BEV center point of any other object and a planned waypoint in $\hat{\boldsymbol{\tau}}$ within the planning horizon T_{plan}. Furthermore, we add a regularizer encouraging $\mathbf{Z}$ to remain likely under the prior, resulting in the following overall cost:

$$\mathcal{C} = -\min_{\substack{n\in\{0,\cdots,N\}\\ t\in\{0,\cdots,T_{\text{plan}}\}}} d(\hat{\boldsymbol{\tau}}^t, \mathbf{s}_n^t) + \lambda \sum_{n=0}^{N_q} \log p_\theta(\mathbf{z}_n \mid \mathbf{S}, \mathcal{M}) \tag{6}$$

where λ controls the strength of the prior regularization.

TABLE I
PERFORMANCE EVALUATION FOR DIFFERENT PEMS. THE COMPARISON OF PEM AND DETECTOR CHARACTERISTICS IS MEASURED IN TERMS OF CUMULATIVE ABSOLUTE DIFFERENCE AREA (CD) FOR DIFFERENT PERCEPTION METRICS. $*$ DENOTES A VARIANT WITHOUT VISIBILITY INPUT, WHILE † DENOTES ADDITIONAL MAP INPUT.

Model	CD-mPrec.	CD-mATE	CD-mAOE	CD-mAVE
Detr3d [2]				
Static Gauss	0.082	0.479	0.453	0.601
MLP + Gauss $*$	0.057	0.116	0.059	0.310
MLP + Gauss	0.061	0.120	0.064	0.204
MLP + Gauss †	0.065	0.162	0.102	0.263
ResNet + Gauss	0.062	0.138	0.068	0.274
ResNet + StudT	0.052	0.117	0.072	0.333
ResNet + StudT †	0.069	0.156	0.078	0.351
OURS	**0.038**	**0.053**	0.060	0.133
w/ KL-Divergence	0.044	0.061	0.066	0.140
w/ $N_q^{fp} = 0$	0.051	0.115	**0.056**	0.182
w/ $N_q^{fp} = 256$	**0.038**	0.055	0.058	**0.128**
BEVFormer [30]				
MLP + Gauss	0.076	0.152	0.076	0.145
ResNet + Studt	0.068	0.142	**0.061**	0.466
OURS	**0.046**	**0.069**	0.069	**0.044**
StreamPETR [3]				
MLP + Gauss	0.066	0.122	0.088	0.089
ResNet + Studt	0.089	0.191	0.100	0.342
OURS	**0.056**	**0.109**	**0.087**	**0.046**

Optimization: Since all components in our framework are differentiable, we approach the optimization problem in (5) via gradient ascent. This results in the following procedure:

1) Infer the prior distribution $p_\theta(\mathbf{z}_n \mid \mathbf{S}, \mathcal{M})$ for the current scene via the corresponding probabilistic encoder, and initialize all $\mathbf{z}_n$ to the mean $\boldsymbol{\mu}^p_{\theta,n}(\mathbf{S}, \mathcal{M})$.
2) Take N_{opt} gradient steps on $\mathbf{Z}$ maximizing the cost function in (6) and clamp all $\mathbf{z}_n$ to the permitted interval $(-\kappa\,\boldsymbol{\sigma}^p_{\theta,n}, \kappa\,\boldsymbol{\sigma}^p_{\theta,n})$ after each update to satisfy the constraint in (5).
3) After each step, determine if the planned trajectory results in a future collision with any object via an intersection check on the BEV bounding-boxes of all ego-object pairs. If no collision is found within N_{opt} iterations, the optimization terminates unsuccessfully.

To generate multiple challenging perception results per scene, we run the above procedure up to N_{trial} times, each time removing the object used to compute the collision cost in (6) in the previous trial from consideration.

IV. EXPERIMENT RESULTS

In this section, we experimentally analyze EMPERROR in terms of (1) its effectiveness in faithfully imitating modern camera-based 3D object detectors and (2) illuminate its utility in probing the robustness of planning to such errors.

Dataset: We utilize the challenging and well-established nuScenes dataset [32] consisting of 1000 real-world sensor logs covering a diverse range of scenarios, each 20 s in length, and tracked 3D annotations of ten different object categories at a frequency of 2 Hz. We use the official detection sub-split of the training set to train the detector and apply it on the tracking sub-split to generate training data for the PEM. We evaluate both on the official validation split.

Target Detectors: To include a variety of error characteristics in our evaluation, we choose three different modern detectors: (1) DETR3D [2] based on sparse object queries, (2) StreamPETR [3] that utilizes temporal object queries that are propagated through time and (3) BEVFormer [30] which utilizes an intermediate temporal BEV-feature grid. All detectors have been trained for 48 epochs, utilizing a ResNet-101 [38] backbone and the official implementations.

A. Evaluation of Error Imitation Quality

Metrics: To measure the perception performance, we follow the official metric definitions of the nuScenes benchmark. These include the Average Precision (AP) as well as regression error metrics for true positives. The latter include the mean Average Translation Error (mATE), mean Average Orientation Error (mAOE) and mean Average Velocity Error (mAVE). For the exact metric definitions, we refer the reader to [32]. However, analyzing the mean precision and error values alone is insufficient, as it does not quantify how the errors evolve with decreasing detection confidence values. To this end, we compare the integral of absolute differences in metric values over all recall intervals between the target detector and PEM, which we term Cumulative Absolute Difference Area (CD).

Baselines: We implement a simple baseline similar to the observation noise used in the Kalman filter, which models the perception error distribution as a scene-independent Gaussian distribution. To this end, we compute the per-class empirical Gaussian over 3D bounding box regression errors and class logits for detections with a ground-truth match on the PEM sub-split of the training dataset. Samples from this distribution are then applied to ground-truth bounding boxes to create noisy perception results. We additionally compute the per-class false-negative rate, at which we randomly drop detections. Since this baseline, which we term Static Gauss, simply models dataset-level statistics, it is incapable of capturing scene-dependent error patterns. Inspired by the PEMs proposed in [15], [13] we also design six object-conditioned baseline configurations that utilize a simple per-object feed-forward network to map the ground-truth state as well as a categorical visibility level [32] to a noisy detection output. Furthermore, we also construct a variant that additionally utilizes the same map encoder as our approach by concatenating the flattened map embedding to the per-object state projections. Note that these methods fail to explicitly model false positives [15], [13] and cannot capture error patterns that depend on other scene elements, such as duplicate detections. Unlike Static Gauss, they can, however, capture correlations between the ground-truth object state and detection errors on a per-object basis. These baselines output the confidence scores for each class, as well as the parameters of a probability distribution for all regression targets. We train the models by optimizing the negative log likelihood for a given target detector. We use the same MLP as EMPERROR for the input state projection, followed by either a three layer MLP utilizing Layer normalization [39] and an ELU activation function [40] or a ResNet [38] as proposed in [13] without dropout. For the probability distribution, we

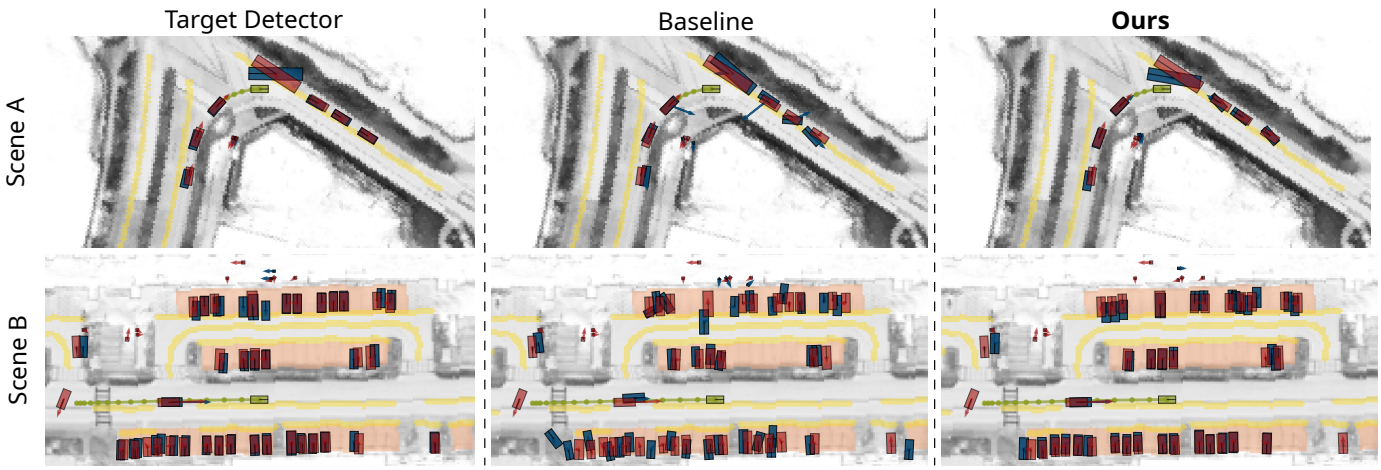
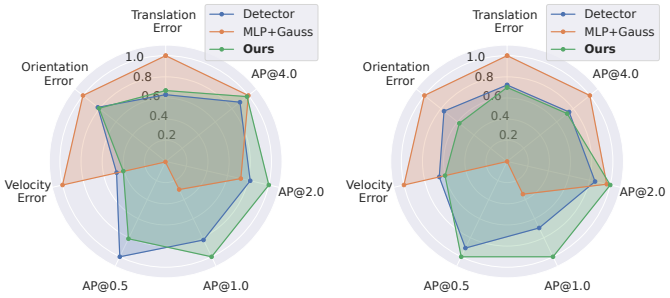


Fig. 3. **PEM Qualitative Results.** Perception errors modeled by a baseline PEM (middle) using an MLP with a Gaussian and EMPERROR (right) compared to DETR3D detections (left). Red boxes indicate ground truth objects, blue boxes the model predictions. While the baseline model samples implausible perception velocities and does not adapt to scene context, such as the parking area in Scene B, our approach closely mimics the target detector.



(a) Car (b) Pedestrian

Fig. 4. **PEM Error Characteristics.** We show the precision and mean regression errors for DETR3D, EMPERROR and the MLP + Gauss configuration as baseline. All metrics are normalized for clearer visual comparison.

either use a multivariate Gaussian as in [15] with a diagonal covariance matrix or a multivariate Student-T distribution as in [13] as the target distribution.

Implementation Details: For EMPERROR, we use $N_q^{gt} = 300$ queries for ground-truth, as well as $N_q^{ft} = 128$ queries for false positives. Each query has a latent dimensionality of $d_h = 256$ while the sampled latent code utilizes $d_z = 32$. This configuration is comparatively lightweight and achieves an inference throughput of roughly 190 scenes per second at batch size 64 on an Nvidia Titan Xp GPU. We choose $\beta = 0.01$ for weighting $\mathcal{L}_{JS}$ after a warm-up period of three epochs, while setting $\alpha = 0.5$ for an equal weighting of forward and backward KL. We filter detector targets and PEM outputs for a minimum confidence score $\max(\mathbf{b}^{cls}) \geq 0.2$ to ignore low confidence predictions.

Results: The performance of our proposed PEM for different detectors is shown in Table I. Compared to both Static Gauss and the object-conditioned baselines, our model more precisely captures the error characteristics of the target detector, leading to consistent improvements in all metrics. For DETR3D [2], which only uses inputs from a single time step, the CD-mPrec. is improved by 26 % over the best baseline. Compared to target detectors that utilize temporal information, our proposed PEM improves the CD-mPrec. for BEVFormer [30] as target by 32 % and by 15 % for StreamPETR [3] respectively. We also show the mean error characteristics for DETR3D in Fig. 4. Especially the reproduction of the translation error, velocity error and average precision is significantly improved in our model compared to the MLP + Gauss baseline.

In Fig. 3, we show two qualitative examples of sampled perception errors for the MLP + Gauss baseline and EMPERROR in comparison to DETR3D [2] as target detector. In the first scene, the baseline model predicts implausible velocity estimates for oncoming traffic, while our approach models this correctly and samples realistic translation errors along line of sight as well reproducing similar orientation errors for large vehicles. The second scene highlights the importance of scene context. In contrast to our model, the baseline fails to sample plausible errors for the orientations of grouped parking vehicles, whilst our approach closely mimics the error modes of the target detector. Additional examples can be found in the supplementary material.

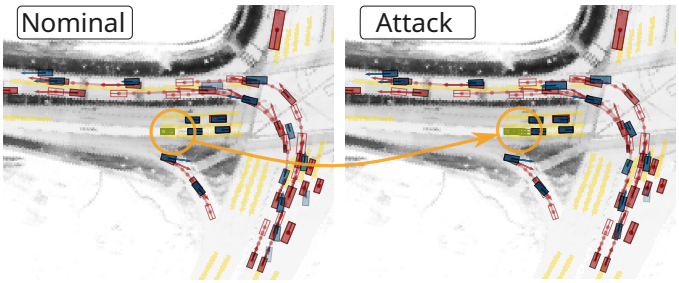
Ablation Study: To verify our key design choices, the *query initialization scheme* and *choice of divergence*, we run experiments varying the number of false positive queries N_q^{fp} and using the standard KL divergence instead of the skew-geometric JS divergence and report the results in Table I. Doubling the number of false-positive queries results in similar performance at a higher computational burden, while omitting them drastically degrades performance, highlighting their importance. When using the KL- instead of the JS^{G_α} divergence we also see degraded performance. Additionally, we observe that it can permit high latent variance, providing an avenue for implausible attacks.

B. Adversarial Perception Errors

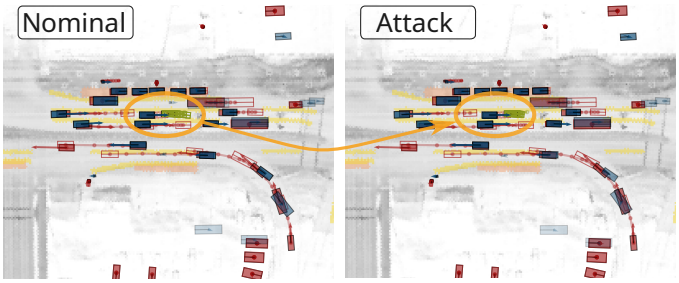
We now apply EMPERROR to probe the robustness of the IL-based planner described Section III-B. To this end, we run our proposed adversarial optimization procedure on scenes from the validation split, which is held-out for all

TABLE II
EFFECTIVENESS OF ADVERSARIAL PERCEPTION ERRORS. LEFT-HAND SIDE: COMPARISON OF BASELINE OPEN-LOOP PLANNING PERFORMANCE. RIGHT-HAND SIDE: ATTAINED INCREASE IN COLLISION RATE (CR) FOR VARYING LATENT SPACE CONSTRAINTS κ.

	Detector			PEM			Adversarial PEM		
	CR	ADE	FDE	CR	ADE	FDE	CR (%)		
Model	(%)	(m)	(m)	(%)	(m)	(m)	$\kappa = 1$	$\kappa = 2$	$\kappa = 3$
Detr3d [2]	3.40	1.23	2.59	3.56	1.22	2.57	4.27 (+20%)	5.09 (+43%)	5.88 (+65%)
BEVFormer [30]	3.20	1.25	2.63	3.36	1.24	2.61	4.47 (+33%)	5.27 (+57%)	6.20 (+85%)
StreamPETR [3]	3.40	1.27	2.67	3.58	1.25	2.60	4.77 (+33%)	5.35 (+49%)	6.28 (+75%)



(a) Not keeping safe distance



(b) Dangerous sudden braking

Fig. 5. **Qualitative Examples of Worst-Case Perception Errors** ($\kappa = 3$)**.** Sampling adversarial perception errors from EMPERROR can induce unwarranted acceleration (Fig. 5a) and sudden braking (Fig. 5b) in the planner, causing collisions. Red indicates ground truth objects, blue the model predictions. Non-filled boxes represent future states.

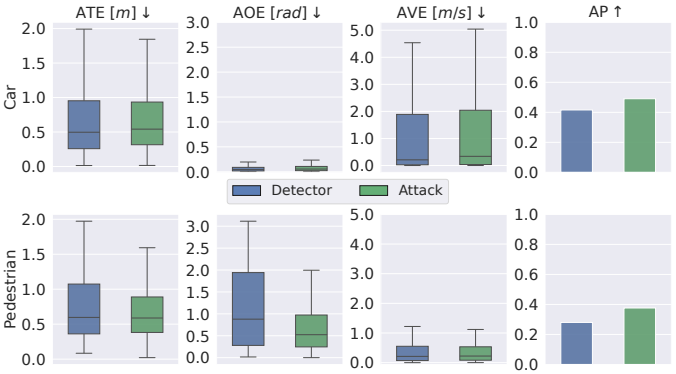


Fig. 6. **Attack Plausibility.** Successful attacks (with $\kappa = 3$) show a similar or more accurate distribution of errors compared to DETR3D, highlighting the realism and conservative nature of our adversarial framework.

involved models during training. To gain an understanding of its baseline performance, we first apply the planner to perception results obtained directly from the target detector. We also verify whether it performs similarly when operating on maximum likelihood samples obtained from EMPERROR (i.e. $\mathbf{z}_n = \boldsymbol{\mu}^p_{\theta,n}(\mathbf{S}, \mathcal{M}) \; \forall \; \mathbf{z}_n \in \mathbf{Z}$). Finally, we optimize for failure over varying latent space constraints κ.

Metrics: As we are interested in inducing unsafe behavior, we use the Collision Rate (CR) as the main metric for this experiment. It measures the percentage of scenes for which the planned trajectory collides with another object within the planning horizon T_{plan}. This is supplemented by the Average Displacement Error (ADE), which measures the average l_2-distance to the human expert trajectory within T_{plan}, as well as the Final Discplacement Error (FDE), which is similar to the ADE but considers only the final waypoint at $t = T_{\text{plan}}$.

Implementation Details: We use the Adam [41] optimizer with a learning rate of $1\text{e}{-1}$ for the adversarial optimization procedure, which we run for a maximum of $N_{\text{opt}} = 100$ iterations per scene, permitting $N_{\text{trial}} = 5$ trials each. The strength of the likelihood regularization is set to $\lambda = 1\text{e}{-1}$. We consider a planning horizon T_{plan} of 3 s and train dedicated versions of the planner on inference results of each target detector using the tracking sub-split.

Results: In addition to the evaluation presented in Section IV-A, the quantitative results reported in Table II further indicate EMPERROR to be a capable PEM in terms of downstream planning. Compared to the actual inference results of the target detector, operating the planner on maximum likelihood proxy samples results in similar a CR, ADE and FDE. Furthermore, when allowing the adversarial optimization procedure to adjust the set of latent variables $\mathbf{Z}$, the baseline CR can be increased by up to 85%, suggesting a critical vulnerability of our prototypical planner to even plausible perturbations in its input. The adversarial optimization most strongly affects the planned trajectory longitudinally, for example by inducing sudden dangerous braking or acceleration while in stationary traffic. We hypothesize this is due to a bias of heading straight-ahead that is common in driving data. This is visualized in Fig. 5. Additional examples can be found in the supplementary material.

Attack Plausibility: Through likelihood regularization Eq. (6), search space constraints Eq. (5) and use of the JS- instead of KL-Divergence Eq. (4), we design our method to be conservative such that it generates plausible attacks rather than extreme perturbations. This is evident from Fig. 6, where we show boxplots of real-world detector errors and EMPERROR samples inducing successful attacks (with $\kappa = 3$) on the same sensor logs: Even after adversarial optimization, EMPERROR produces results that either match the detector

distribution or show lower errors, both in terms of the median and spread of the distribution, highlighting their plausibility.

V. Conclusion

We presented EMPERROR, a novel generative perception error model (PEM) that mimics the outputs of a perception system given a ground-truth scene representation. We have demonstrated its utility as an evaluation tool by probing an imitation learning (IL)-based planner in terms of its robustness to noise in its inputs. Furthermore, we showed that EMPERROR more accurately captures the target detector's error characteristics than PEMs used in previous work. However, there are remaining limitations opening avenues for improvement in future work. Firstly, we manually designed a cost function to induce a specific failure mode (i.e. collision). To enable inducing a wider range of planning failures, learning a general driving cost function from data [42] is an interesting direction. Secondly, extending EMPERROR to model latent features instead of explicit intermediate representations, such as 3D bounding boxes, would enable stress-testing modular end-to-end trainable architectures, which have recently gained popularity [35], [36].

References

[1] J. Janai, F. Güney, A. Behl, and A. Geiger, *Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art*. Foundations and Trends in Computer Graphics and Vision, 2020, vol. 12, no. 1-3.

[2] Y. Wang, V. . Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. M. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," in *CoRL*, 2021.

[3] S. Wang, Y. Liu, T. Wang, Y. Li, and X. Zhang, "Exploring object-centric temporal modeling for efficient multi-view 3d object detection," in *ICCV*, 2023.

[4] T. Li, L. Chen, H. Wang, Y. Li, J. Yang, X. Geng, S. Jiang, Y. Wang, H. Xu, C. Xu, J. Yan, P. Luo, and H. Li, "Graph-based topology reasoning for driving scenes," *arXiv.org*, vol. 2304.05277, 2023.

[5] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *ECCV*, 2020.

[6] A. Hu, Z. Murez, N. Mohan, S. Dudas, J. Hawke, V. Badrinarayanan, R. Cipolla, and A. Kendall, "FIERY: Future instance segmentation in bird's-eye view from surround monocular cameras," in *ICCV*, 2021.

[7] K. Renz, K. Chitta, O.-B. Mercea, A. S. Koepke, Z. Akata, and A. Geiger, "Plant: Explainable planning transformers via object-level representations," in *CoRL*, 2022.

[8] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in *CoRL*, 2023.

[9] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," in *CVPR Workshops*, 2021.

[10] N. Montali, J. Lambert, P. Mougin, A. Kuefler, N. Rhinehart, M. Li, C. Gulino, T. Emrich, Z. Yang, S. Whiteson, B. White, and D. Anguelov, "The waymo open sim agents challenge," *arXiv.org*, vol. 2305.12032, 2023.

[11] A. Filos, P. Tigas, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal, "Can autonomous vehicles identify, recover from, and adapt to distribution shifts?" in *ICML*, 2020.

[12] C. Innes and S. Ramamoorthy, "Testing rare downstream safety violations via upstream adaptive sampling of perception error models," in *ICRA*, 2023.

[13] J. Sadeghi, N. A. Lord, J. Redford, and R. Mueller, "Attacking motion planners using adversarial perception errors," *arXiv.org*, vol. 2311.12722, 2023.

[14] A. Piazzoni, J. Cherian, M. Slavik, and J. Dauwels, "Modeling perception errors towards robust decision making in autonomous vehicles," in *IJCAI*, 2021.

[15] J. Sadeghi, B. Rogers, J. Gunn, T. Saunders, S. Samangooei, P. K. Dokania, and J. Redford, "A step towards efficient evaluation of complex perception tasks in simulation," in *NeurIPS Workshops*, 2021.

[16] P. Mitra, A. Choudhury, V. R. Aparow, G. Kulandaivelu, and J. Dauwels, "Towards modeling of perception errors in autonomous vehicles," in *ITSC*, 2018.

[17] A. Piazzoni, J. Cherian, J. Dauwels, and L.-P. Chau, "On the simulation of perception errors in autonomous vehicles," *arXiv.org*, vol. 2302.11919, 2023.

[18] K. Wong, Q. Zhang, M. Liang, B. Yang, R. Liao, A. Sadat, and R. Urtasun, "Testing the safety of self-driving vehicles by simulating perception and prediction," in *ECCV*, 2020.

[19] X. Ju, Y. Sun, Y. Hao, Y. Li, Y. Qiao, and H. Li, "Perception imitation: Towards synthesis-free simulator for autonomous vehicles," *arXiv.org*, vol. 2304.09365, 2023.

[20] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *CVPR*, 2018.

[21] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," *CVPR*, 2021.

[22] W. Ding, M. Xu, and D. Zhao, "Learning to collide: An adaptive safety-critical scenarios generating method," in *IROS*, 2020.

[23] S. Suo, K. Wong, J. Xu, J. Tu, A. Cui, S. Casas, and R. Urtasun, "Mixsim: A hierarchical framework for mixed reality traffic simulation," in *CVPR*, 2023.

[24] J. Wang, A. Pun, J. Tu, S. Manivasagam, A. Sadat, S. Casas, M. Ren, and R. Urtasun, "Advsim: Generating safety-critical scenarios for self-driving vehicles," in *CVPR*, 2021.

[25] D. Rempe, J. Philion, L. J. Guibas, S. Fidler, and O. Litany, "Generating useful accident-prone driving scenarios via a learned traffic prior," in *CVPR*, 2022.

[26] N. Hanselmann, K. Renz, K. Chitta, A. Bhattacharyya, and A. Geiger, "KING: Generating safety-critical driving scenarios for robust imitation via kinematics gradients," in *ECCV*, 2022.

[27] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.

[28] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *NIPS*, 2015.

[29] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *ICLR*, 2014.

[30] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *ECCV*, 2022.

[31] S. Suo, S. Regalado, S. Casas, and R. Urtasun, "Trafficsim: Learning to simulate realistic multi-agent behaviors," in *CVPR*, 2021.

[32] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv.org*, 2019.

[33] J. Deasy, N. Simidjievski, and P. Liò, "Constraining variational inference with geometric jensen-shannon divergence," in *NeurIPS*, 2020.

[34] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *ICLR*, 2017.

[35] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *CVPR*, 2023.

[36] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," *ICCV*, 2023.

[37] S. Doll, N. Hanselmann, L. Schneider, R. Schulz, M. Cordts, M. Enzweiler, and H. P. Lensch, "Dualad: Disentangling the dynamic and static world for end-to-end driving," in *CVPR*, 2024.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[39] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv.org*, vol. 1607.06450, 2016.

[40] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv.org*, vol. 1511.07289, 2015.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[42] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *AI*, vol. 297, p. 103500, 2021.