Better and safer autonomous driving with predicted object relevance

Andrea Ceccarelli
University of Florence
Florence, Italy
andrea.ceccarelli@unifi.it

Leonardo Montecchi
NTNU
Trondheim, Norway
leonardo.montecchi@ntnu.no

***Abstract*—Object detection in autonomous driving consists in perceiving and locating instances of objects in multi-dimensional data, such as images or lidar scans. Very recently, multiple works are proposing to evaluate object detectors by measuring their ability to detect the objects that are most likely to interfere with the driving task. Detectors are then ranked according to their ability to detect objects that are relevant, rather than the general accuracy of detection. However, there is little evidence so far that isolating the most relevant objects may contribute to improvements in the safety and effectiveness of the driving task. This paper defines and exercises a strategy to i) set-up and deploy object detectors that successfully extract knowledge on object relevance, and ii) use such knowledge to improve the trajectory planning task. We show that, given the output of an object detector, filtering objects based on their predicted relevance, in combination with the usual confidence threshold, improves the quality of trajectories produced by the downstream trajectory planner. We conclude the paper showing that information on object relevance should be further exploited and we sketch some directions for future work.**

***Index Terms*—Autonomous driving, object detection, trajectory planning, safety, object relevance, object criticality**

I. Introduction

Object detection is a fundamental task for autonomous pipelines, which require object detection as part of their perceptual interface to the environment [1]. Noteworthy, in the autonomous driving domain, under the name of object detectors are actually included perception models that go beyond the mere identification of object location and classification, but that instead also identify additional attributes such as object size, distance from the observer, orientation, and velocity [2].

Autonomous driving pipelines (as opposed to end-to-end autonomous driving [3]) provide several advantages, but they generally incur the disadvantage that individual elements of the pipeline do not optimize for system-wide or downstream performance metrics [4]. For example, several initiatives aim to perform safe and robust trajectory planning based on the inputs acquired from an object detector, but under the assumption that the object detector provides a sufficiently accurate representation of the scene. However, most object detectors are general-purpose components, whose results are produced without considering the downstream tasks.

To overcome this problem, we argue that the steps of the pipeline beyond object detection should consider the relevance of detected objects, in addition to the plain list of detections that is typically provided by an object detector. Adding information on the relevance of objects to the successive pipeline tasks, and especially to trajectory planning, may allow improving the overall safety and effectiveness of the driving task. In this paper, we use the term *relevance* of an object to mean its importance to a certain downstream task. In the context of autonomous driving, criticality (e.g., distance of an object to the ego vehicle) is a concrete example of object relevance for the trajectory planning task.

Very recently, some works have proposed to evaluate object detectors based on their ability to detect objects that are most relevant to the scene, rather than based on their ability to detect as many objects as possible, and as accurately as possible [5], [6], [7], [8]. In this paper we move a step forward: we propose that reasoning on object relevance is useful not only for selecting the most suitable object detector, but it is of fundamental support at runtime, in the successive steps of the pipeline, to prioritize which objects to consider when detection is uncertain.

More specifically, we investigate the impact of two approaches for filtering the output of object detectors taking object relevance into account. To evaluate these two strategies, we observe the performance of a trajectory planner when different sets of predicted objects are selected. After defining our problem and the proposed methodology, we experiment with six state-of-the-art object detectors on the nuScenes [9] dataset. As a measure of trajectory quality, we rely on the planning KL-divergence metric [10] from the literature, which measures the distance of the computed trajectory from a ground truth reference.

The main contributions of our work are:

1) an approach to make the relevance of an object an input to the planning task, to ultimately improve the effectiveness and safety of the driving task. This contrasts with the majority of works on object relevance, which exploit the concept only for selecting the most suitable object detector. In our proposal, the estimation of object relevance is no longer used to assess object detectors only, but it is a step of the autonomous driving pipeline.
2) experimental results showing that, while predicted objects are traditionally selected based on a confidence threshold only, more flexible approaches exist to decide on the inclusion of a predicted object, and they may be

more convenient for trajectory planning. Such approaches should include a combination of multiple parameters (in our paper, predicted relevance and detection confidence).
3) evidence that further research is needed in this direction, as more advanced ways to exploit relevance information may further improve the quality of downstream tasks.

II. Related Works and Advancements

In the literature, numerous works have focused on building better and safer autonomous driving systems [11]. Among the multitude of research activities [2], we concentrate on related works that i) measure the object relevance in the context of autonomous driving, and ii) exploit this information in the later steps of the pipeline, for safe and effective trajectory planning.

Measuring object relevance. In the last four years, context-aware and safety-aware metrics for object detection have been proposed, to evaluate object detectors with respect to the safety and reliability of the system in which they will operate.

Lyssenko et al. [5] measure the maximum distance at which pedestrian detection does not fail, while Wolf et al. [6] weight detections according to the position and estimated time-to-collision with the object. Volk et al. [7] propose a detection metric that includes the relevance of predicted objects with respect to the observer, Ceccarelli et al. [8] associate a criticality to each object based on distance from the observer and the estimated velocity, and Topan et al. [12] present a model to compute safety zones and define safety evaluation metrics for analyzing perception performance of an autonomous vehicle. Further, Liao et al. [13] propose a safety metric, especially for 3D object detectors in autonomous driving contexts, by combining an Intersection-over-Ground-Truth (IoGT) measure and a distance ratio.

The effects of object detection in the pipeline. Some authors propose to consider the whole system level or the effect of misdetection in the entire pipeline, rather than specifically to assess the object detector. McAllister et al. [4] propose an approach that considers how predictions will be used downstream to improve accuracy whenever prediction errors would cause a large change in control outputs. Similarly, but focusing on security and adversarial attacks, Wang et al. [14] question whether previous works can achieve system-level effects (e.g., vehicle collisions, traffic rule violations) under real autonomous driving settings. They then study the effect of adversarial attacks on the whole driving task, resulting in system-level effects rather than just a misclassification.

While the impact of object detection on the planning task is well understood, only very recently there have been efforts to quantify it with metrics. Most notably, the work in [15] and [16] evaluates the impact of object detection on driving performance, with the aim to propose metrics for comparing object detectors. The best detection models should make the planner compute a trajectory as close as possible to the one computed using ground truth information [16].

Differences from previous works. In this paper, we bring forward the position that it is necessary to exploit the predicted object relevance to improve trajectory planning, as a crucial step of the autonomous driving pipeline. Most of the reviewed works do not provide evidence that their solution actually increases the safety and effectiveness of the planning task. We demonstrate that object detection should be studied not just to maximize detection accuracy, but for the higher-level purpose of safety and effectiveness: for this purpose, additional information as object relevance needs to be produced during the object detection process, and a deeper integration with the trajectory planner is required in the evaluation of detectors.

III. Background

A. Object Detection

The task of object detection consists in locating and classifying semantic objects of certain object classes within an input *sample*, with the sample consisting of visual images acquired from visual cameras, or 3D point clouds from lidar scans. The output of an object detection model is a list of predicted *bounding boxes* (BBs), with *confidence scores* and *labels* [17]. BBs are tightly-bound boxes encompassing objects in the sample, represented in 2D and 3D as rectangles and cuboids, respectively. The confidence score reflects the confidence of the detection model in each predicted BB. The label is the predicted semantic class of each BB.

Noteworthy, models developed for autonomous driving tasks typically do not stop at identifying BBs, but they also determine the kind of object (i.e., they perform classification) and further, they compute key attributes like orientation, velocity, and distance from the observer.

B. Metrics to evaluate object detectors

To evaluate the predictions produced by an object detector, a comparison between the predicted BBs and the ground truth BBs is performed. In practice, an object detector predicts many BBs, each with a confidence score in the interval $[0, 1]$. Typically, most of the BBs in the raw output of an object detector have a low confidence score, and some of them may refer to the same ground truth object. To retain only the most credible bounding boxes, a *confidence threshold* θ is established as a configuration parameter: all the BBs with a confidence score above θ are then considered as actual *predictions*, while the others are discarded.

Once BBs have been filtered, the identification of true positives (TPs), false positives (FPs), and false negatives (FNs), is based on some definition of distance between the predicted BBs and the ground truth BBs. Typically, the distance between center points is used [9]. If the distance is below a *distance limit*, it is considered a correct detection (true positive, TP). If no predicted BB matches the distance limit, then the object is not detected and it counts as a false negative (FN). Predicted bounding boxes that are farther than the distance limit from all ground truth bounding boxes are considered false positives (FPs). True negatives (TNs) are not taken into account, because there are infinite BBs that should not be detected within any given image [18].

Several aggregated measures based on TPs, FPs, and FNs, can summarize the performance of object detectors; most typically, precision, recall, and average precision are used. *Precision*, $\mathrm{P} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$, indicates how many of the selected items are relevant; conversely, *Recall*, $\mathrm{R} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$, indicates how many items from the ground truth are correctly selected. Computing precision and recall for varying confidence thresholds, results in the precision-recall curve. This curve offers a graphical summarizing view of the precision-recall tradeoff.

Average Precision (AP), first presented in [19], is currently deemed the most suitable measure to compute and rank the performance of object detectors [20]. AP summarizes the precision-recall curve as the weighted mean of precision scores achieved at different confidence thresholds. More precisely, $AP = \sum_n (R_n - R_{n-1}) P_n$, where P_n and R_n are the precision and recall at the n-th confidence threshold. The mean Average Precision (mAP) is used to summarize the AP obtained on different classes of objects and different distance limits.

C. Object Criticality Model

While the above metrics indicate the ability of object detectors to accurately predict instances of objects in a scene, they do not consider the relevance of such objects within the specific scenario. Research on object detection in safety-critical environments has raised the problem of defining safety-aware evaluation metrics. In this paper, we use object relevance as defined by the Object Criticality Model (OCM, [8]).

In the OCM, a criticality value is assigned to each object in a specific scene, based on safety-relevant factors relating the object and the ego vehicle. Three factors are considered for computing the criticality of an object B: distance, colliding trajectory, and time-to-collision, which result in three individual criticality scores, $\kappa_D(B)$, $\kappa_R(B)$, and $\kappa_T(B)$. Each of these scores ranges in the interval $[0, 1]$, with 1 meaning maximum criticality. The overall criticality of an object, $\kappa(B)$, is then defined as a combination of the three above scores. Criticality values can be computed for any object B, being it a ground truth or a predicted object. In the former case, the computation uses ground truth information on position and velocity (resulting in ground truth criticality); in the latter case, the estimated position and velocity are used (resulting in predicted criticality).

The model depends on three parameters, D, R, and T, which define a threshold after which the corresponding criticalities $\kappa_d(B)$, $\kappa_r(B)$, and $\kappa_t(B)$ assume value 0. For example, $D = 30$ means that for objects farther than 30 meters $\kappa_d(B) = 0$. In the same way, the triple $D, R, T = 15, 20, 6$ means that criticality is assigned from 0 to 1 to objects that are estimated to have at least one of the following characteristics: i) being closer than 15 meters from the ego vehicle; ii) being on a potential colliding trajectory within 20 meters from the ego vehicle; or iii) approaching a potential collision point within 6 seconds.

The performance of a detector is then measured in terms of "how much criticality" it can detect, comparing the ground truth criticalities against the predicted criticalities. More specifically, the authors of [8] define two metrics called *reliability-weighted precision* ($P_{\mathcal{R}}$), and *safety-weighted recall* ($R_{\mathcal{S}}$), as variants of the Precision and Recall metrics in which objects are weighted based on their criticality score. Further, the *Average Critical Precision* ($\mathrm{AP}_{\mathrm{crit}}$) is a variant of AP, computed using $P_{\mathcal{R}}$ and $R_{\mathcal{S}}$. Consequently, a mean Average Critical Precision $\mathrm{mAP}_{\mathrm{crit}}$ can also be computed.

D. Planning KL-divergence

In [16], Philion et al. argue that the evaluation of the performance of perception systems in autonomous vehicles should be aligned with the downstream task of trajectory planning. Planning is a crucial part of the autonomous pipeline, so the "best" detection models should be those that make the planner compute a trajectory as close as possible to the one computed on ground truth information. Based on this observation, they propose the *Planning KL-divergence* (pkl) metric, as a measure of the difference between the trajectory planned based on ground truth objects, and the trajectory planned based on the output of an object detector. Being a measure of divergence, a perfect detection would receive a pkl score of 0, corresponding to no divergence between the trajectories. Further details can be found in [16].

In this paper we use pkl as an estimate of the quality of the planned trajectory, and we investigate how different strategies for filtering objects impact on such metric. Our hypothesis is that prioritizing object relevance improves the planned trajectory.

IV. Technical Environment

A. The nuScenes Dataset

Recent years have seen the release of several sophisticated datasets which have played important roles in the advancement of 3D object detectors in autonomous driving. For our work, we used nuScenes [9], as it is the one that was used by the original works on pkl [16] and OCM [8].

nuScenes [9] was released in 2019 as a multimodal dataset for the task of training and evaluating perception systems for autonomous driving. nuScenes comprises 1000 driving scenes of 20 seconds each, acquired in Boston and Singapore, under a wide array of situations and environmental conditions. Samples are collected at 2Hz frequency, for a total of 40 samples per scene. Highly accurate annotations of objects from 23 classes are provided, including the semantic category, bounding boxes, and attributes like speed, coordinates, visibility (line of sight with the ego vehicle), and orientation.

The full nuScenes dataset is split into three parts, namely the training, validation, and test sets, consisting of 700, 150, and 150 scenes, respectively. Annotations are only provided for the training and validation sets, as the test set is utilized for scoring online submissions to the nuScenes challenges on detection [9].

B. The MMDetection3D Toolbox

MMDetection3D [21] is a part of the open-source object detection toolbox MMDetection [22], implementing a large set of detection methods and components related to 3D object detection. MMDetection3D provides specific integration with the nuScenes dataset, which makes it particularly suited for our work.

We selected six different pre-trained models from the MMDetection3D *model zoo* [22], briefly described in the following. The links to the trained models, with details on weights and configuration parameters, are available in our repository [23].

POINTP: PointPillars [24] was proposed as an encoder that learns features from vertical columns (pillars) of point clouds resulting from lidar scans. The PointPillars architecture consists of three stages: a feature encoder network, to transform a 3D point cloud into a pseudo-image; a 2D convolutional backbone, for extracting high-level features from the pseudo-image; and a detection head for BB classification and regression. We use the PointPillars model combined with Feature Pyramid Network (FPN, [25]), which generates a pyramid of feature maps [26].

SECFPN: The model relies on the SECOND (Sparsely embedded convolutional detection, [27]) backbone, combined also in this case with FPN. Also this model uses point clouds.

SSN: The model relies on the shape-aware grouping heads used in the Shape Signature Networks (SSN, [28]). SSN is presented as a novel solution for shape encoding; its shape-aware grouping heads bring objects with similar shapes together, to share weights based on the object size, e.g., the bus and truck classes need a heavier head than the car class. We used the PointPillars model combined with the SSN shape-aware grouping heads.

FCOS3D: FCOS3D [29] uses visual cameras only. The backbone is a pre-trained ResNet101 with deformable convolutions. The neck is the Feature Pyramid Network (FPN), which generates a pyramid of feature maps. Finally, the head that produces the final predictions (object class, location, etc.) relies on a strategy similar to RetinaNet, which applies shared heads to operate the detection of multiple targets.

PGD: PGD [30] is another approach that relies on visual cameras. PGD is a simple yet effective monocular 3D detector. It enhances the FCOS3D model by involving local geometric constraints and improving instance depth estimation.

REG: The implementation of the RegNetX model from [31] is based on PointPillars. The model is the resultant of a methodology to produce a low-dimensional design space consisting of simple, regular networks called RegNet.

C. Developed code and experimental setup

The experimental work presented in this paper exploits the *nuScenes devkit* [32] and the pkl library [33], which have been customized for this work. The nuScenes devkit implements an API for parsing and loading data from nuScenes, and functionality related to object detection. We extended the library with the ability to measure the estimated criticalities of predicted objects, and to attach such information to the returned objects. The pkl library includes the trajectory planner already pre-trained for nuScenes, together with the code to compute pkl in [33]. We customized the library to facilitate the integration with our code, and to customize the visualization of results.

The modified libraries, together with the code and instructions to reproduce results, are available at [23]. All the experiments have been executed on a server with Intel(R) Core (TM) i5-8350U, 192GB RAM, and NVIDIA Quadro RTX 6000 GPU. The generation of results required above 20 full days of computation.

V. Methodology

A. Research Questions

To understand whether object relevance improves the effectiveness of the planning task, we discuss and experimentally explore the following two research questions.

RQ1: Can we improve the effectiveness of trajectory planning by filtering objects based on predicted relevance?: We argue that filtering out BBs based on the predicted relevance, i.e., removing predicted objects that may be deemed not relevant for the planning task, has a positive impact on the trajectory planner. Our hypothesis is that the reduction of unnecessary elements avoids creating unnecessary confusion to the trajectory planner, and consequently, reduces unnecessary actions such as braking or steering. Note that the filtering in this case occurs after objects have been filtered for detection confidence.

RQ2: Can we further improve trajectory safety and effectiveness by exploiting information on object relevance?: The above approach may neglect safety issues, because objects that are detected with a low confidence score but that may be relevant for the driving task (high predicted relevance) would be excluded. Including all the objects with predicted relevance above a given threshold, independently of the confidence score, would mitigate this issue, but at the cost of additional false positives and thus potentially a reduction of the effectiveness of the driving task. We want to investigate whether it is possible to further the planned trajectory by better exploiting relevance information. In particular, we aim at improving trajectory safety while still retaining its effectiveness.

B. Methodological approach

As mentioned earlier, we use *criticality* as defined by the OCM as a measure of object relevance. For this reason, we will use the term criticality in the rest of the paper.

1) Addressing RQ1: After the initial filtering on prediction confidence, we establish a criticality threshold κ and we filter out predicted objects that have a value of predicted criticality lower than a criticality threshold κ. That is, we select predictions based on both a confidence threshold θ and a criticality threshold κ. Then, the pkl is computed for each sample in the validation set, using the predictions that are retained after filtering. Note that setting different θ and κ thresholds results in different pkl values.

Then, we compare the *best pkl value* obtained with this strategy (across all threshold values) with the pkl baseline, that is, the best value obtained using the optimal confidence

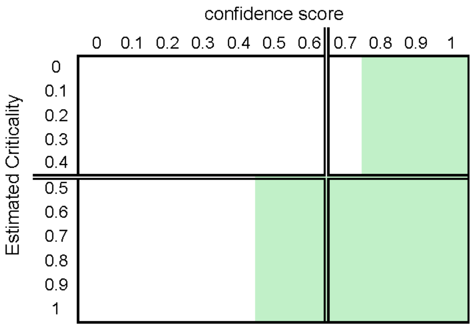


Fig. 1. The colored area represents predicted BBs that are kept, as opposed to using a single confidence threshold $\theta = 0.65$ (vertical line) or a single criticality threshold $\kappa = 0.45$ (horizontal line).

threshold. Lower values (better pkl) would mean that discarding objects that are estimated as less relevant to the scene, as perceived by the object detector, improves the quality of the trajectory.

2) Addressing RQ2: While there can be different ways to combine criticality and detection confidence, in this paper we aim to show that further improvement is possible, and that research in this direction is needed. Therefore, we the aim to find a trade-off between confidence score and predicted criticality that enables further improvement on the pkl score.

We define a confidence threshold θ, a criticality threshold κ, and a correction factor λ. Then, given a predicted BB with confidence score θ_s and predicted criticality κ_s, the BB is included and used for trajectory planning if one of the following two conditions hold:

$$\theta_s > \theta \cdot \lambda \tag{1}$$

$$\kappa_s > \kappa \text{ and } \theta_s > \theta / \lambda \tag{2}$$

The correction factor λ increases the confidence requirement for objects with low criticality (1), and reduces it for objects with high criticality (2). That is, if an object is estimated to have high criticality, it is included even if we the confidence in the detection is low. In other words, we decide if maintaining or discarding a prediction based on a bi-dimensional space, rather than on a single dimension. This is graphically represented in Figure 1, for the case $\theta = 0.65$, $\kappa = 0.45$, and $\lambda = 1.3$. These are representative and plausible values, as will be evident from the results in Section VI.

C. Practical setup and application of the methodology

To realize the above methodology we proceed as follows.

a) Download and validation of detectors: First, the six object detectors described in Section IV-B are downloaded, and they are executed against the nuScenes validation set of 150 scenes. Results are compared with the MMDetection3d documentation to verify that the object detectors are working as expected.

b) Establishing of a reduced validation set: We randomly select 10 scenes out of the 150 scenes of the validation set. This step is due to performance issues. In particular, the computation of a single pkl value is already very computationally expensive, and we need to compute multiple pkl values under different configurations. Applying the entire methodology under multiple configurations, on the entire validation set, and for the whole set of object detectors, would be exceedingly time-consuming, in the order of months of execution. We refer to the 10 selected scenes as the reduced validation set. Note that each scene still contains 40 annotated samples.

TABLE I
PERFORMANCE OF THE OBJECT DETECTORS, FOLLOWING THE NUSCENES EVALUATION LIBRARY [32], AND THE OCM FROM [8].

Model	mAP 150 scenes	mAP 10 scenes	mAP$_{\text{crit}}$ 10 scenes	Best D, R, T
FCOS3D	0.32	0.32	0.407	25, 5, 4
PGD	0.33	0.35	0.40	20, 10, 16
POINTP	0.35	0.35	0.43	25, 5, 4
REG	0.44	0.45	0.53	25, 5, 4
SECFPN	0.35	0.35	0.419	25, 5,4
SSN	0.46	0.46	0.54	25, 5, 4

(a) Results for all the objects

Model	mAP 150 scenes	mAP 10 scenes	mAP$_{\text{crit}}$ 10 scenes	Best D, R, T
FCOS3D	0.49	0.49	0.65	20, 5, 4
PGD	0.53	0.53	0.68	20, 5, 4
POINTP	0.78	0.78	0.89	25, 5, 4
REG	0.81	0.81	0.90	25, 5, 4
SECFPN	0.81	0.81	0.90	25, 5, 4
SSN	0.83	0.83	0.91	25, 5, 4

(b) Results for cars

c) Collection of raw BB outputs from all the detectors: The object detectors are exercised against the reduced validation set. This allows collecting, for each object detector, a json file that describes all the predicted BBs, independently of their confidence scores, for each nuScenes sample. This list is not filtered according to any confidence threshold.

From this, we compute the Average Precision (AP) of the six object detectors for each of the 23 object classes, as well as the mAP, which is the mean of the average precision amongst all classes and amongst the different distance limits. The nuScenes official evaluation library uses four distance limits, namely 0.5, 1.0, 2.0, and 4.0 meters.

d) Representativeness of the reduced validation set: Table I(a) reports the mAP on the reduced validation set and on the 150 scenes of the full validation set. We observe that values are very similar, meaning that the 10 scenes we have selected are a good representation of the entire validation set. We also compute the mean Average Critical Precision (mAP$_{\text{crit}}$), with results that are compatible with those from [8]; for the mAP$_{\text{crit}}$, we report the D, R, T parameters that lead to the highest scores. Last, we repeat the investigation when only car BBs are considered, as these are the most relevant object in the nuScenes dataset, both in terms of frequency (an average of 20 cars per sample) and of semantics, as it is a dataset of a vehicle driving in an urban area. Results are in Table I(b).

Next, the following set of operations is repeated twice: first considering the whole set of objects, and then when only car

TABLE II

BEST MEAN, MEDIAN AND MAXIMUM PKL, AND CONFIDENCE THRESHOLD θ, FOR THE SIX OBJECT DETECTORS. BASE VALUES ARE OBTAINED WHEN USING THE CONFIDENCE THRESHOLD ONLY.

Model	Mean pkl	θ	Median pkl	θ	Maximum pkl	θ
FCOS3D	4.10	0.15	0.99	0.25	98.92	0.25
PGD	4.78	0.05	1.03	0.05	116.46	0.05
POINTP	78.33	0.55	24.86	0.55	377.53	0.45
REG	78.95	0.55	22.75	0.55	377.47	0.40
SECFPN	65.56	0.50	16.49	0.50	372.93	0.40
SSN	3.49	0.25	0.71	0.30	118.75	0.30

(a) Results when all objects are considered

Model	Mean pkl	θ	Median pkl	θ	Maximum pkl	θ
FCOS3D	2.12	0.15	0.59	0.20	67.66	0.15
PGD	2.72	0.10	0.70	0.10	116.46	0.05
POINTP	74.91	0.55	23.67	0.55	322.71	0.55
REG	76.00	0.55	23.62	0.45	322.32	0.45
SECFPN	60.52	0.55	9.44	0.45	321.96	0.45
SSN	2.30	0.35	0.51	0.35	89.88	0.25

(b) Results when only car objects are considered

BBs are considered.

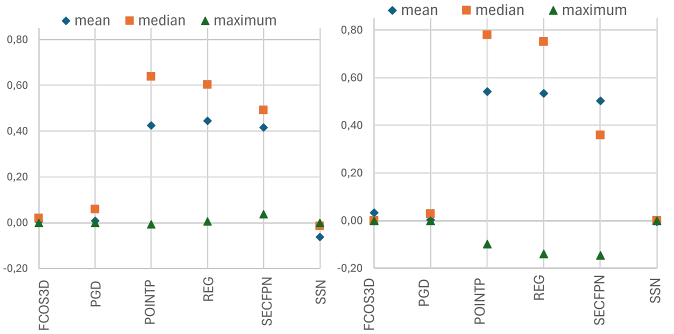
e) Computation of the pkl baseline (using only the confidence threshold): We compute the pkl for the six object detectors. The process to compute pkl using the library provided by its authors [33] is the following. First, a confidence threshold is set, and only objects predicted with a confidence score greater than that are retained and fed to the pkl library. The library will use these predicted objects to predict a trajectory, and to measure its distance from the ground truth trajectory. To find the best achievable pkl values, this process must be repeated with different confidence thresholds: we repeatedly exercise the pkl library using confidence thresholds θ in the range $]0, 1[$, with steps of granularity of 0.05. As a last note, according to [10], the pkl should not be evaluated just as a single value, but it is measured in terms of mean, median, and maximum pkl, computed on all the samples of the validation set.

The six object detectors, the best obtained mean, median, and maximum pkl values, and the confidence threshold θ used to obtain such values are reported in Table II. We remind that the closest pkl is to 0, the better. As already observed in other works, pkl and mAP are quite unrelated; for example, REG shows an excellent mAP but also the worst pkl. Note that the optimal confidence threshold is strictly dependent on the model: values of θ referring to different models should not be compared.

This provides us with the basic information to start the exploration of the two targets previously defined.

f) Exploring the two research questions: The practical approach consists of repeating the steps for the computation of the pkl just described above, but filtering objects based on the different criteria, that is: i) considering the criticality threshold κ and the confidence threshold θ; and ii) adding a third parameter, the correction factor λ.

We consider multiple combinations of κ, θ and λ. Values of κ and θ range in $]0, 1[$, with steps of 0.05, while the values tested for λ range in $]1.0, 1.6]$, with steps of 0.1.



(a) All objects

(b) Car only

Fig. 2. Percentage improvement of pkl metric when objects with low criticality are excluded from the trajectory planning task.

Given that the predicted criticality depends on three values D, R, T, as explained in Section III-C, proper experimentation requires testing multiple combinations of these values, in addition to the combinations of θ, κ, and λ. Only in this way we can really identify the best pkl value that can be achieved by taking into account criticality. The D, R, T combinations are represented in a grid search $D_{candidate} \times R_{candidate} \times T_{candidate}$, such that: $D_{candidate} = \{5, 10, \ldots, 45, 50\}$ meters; $R_{candidate} = \{5, 10, \ldots, 45, 50\}$ meters; and $T_{candidate} = \{4, 8, \ldots, 36, 40\}$ seconds. The values of D and R are set in the established ranges because the nuScenes object detection challenge only includes objects closer to 50 meters; the range of T is from [8], where it is practically shown that the contribution of T is irrelevant beyond $T = 40$ seconds.

To avoid measuring an exceeding number of combinations of D, R, T, κ, θ, λ, and also in light of the computational time required to measure the pkl, we sample configurations from the whole space of combinations following a uniform distribution, and then we investigate the area where we get the most promising results.

VI. RESULTS

A. RQ1: Improvement when filtering for criticality

We compare the baseline pkl (reported in Table II), and the best pkl values obtained when we filter out objects with low predicted criticality (below a threshold κ).

Figure 2 reports the percentage improvement in pkl, computed as 1 minus the ratio between the pkl from Table II, and the pkl computed after filtering out objects as described earlier. The improvement is substantial for the *mean* and *median* pkl of POINTP, REG, and SECFPN. Instead, it is very small for FCOS3D and PGD, and in two cases negative for SSN (the mean and median pkl when considering all the objects). Noteworthy, FCOS, PGD, and SSN had already a very low (i.e., good) baseline pkl, so even small improvements can result in appreciable improvements.

Instead, for some cases, the best *maximum* pkl increases (i.e., it gets worse). The maximum pkl represents the worst case among the samples in the validation set, in which the

TABLE III
PARAMETERS USED TO MEASURE THE MEAN, MEDIAN, AND MAXIMUM PKL FOR THE EFFECTIVENESS IMPROVEMENT TARGET.

	Mean pkl			Median pkl			Maximum pkl		
Model	θ	κ	D, R, T	θ	κ	D, R, T	θ	κ	D, R, T
FCOS3D	0.15	0.15	50, 10, 24	0.25	0.30	50, 10, 24	0.25	0.30	50, 10, 24
PGD	0.05	0.15	50, 50, 24	0.05	0.20	50, 50, 24	0.05	0.20	50, 50, 24
POINTP	0.55	0.65	5, 5, 4	0.55	0.65	5, 5, 4	0.45	0.65	15, 10, 20
REG	0.55	0.60	5, 5, 4	0.5	0.65	5, 5, 4	0.4	0.35	15, 5, 4
SECFPN	0.4	0.65	15, 5, 4	0.4	0.65	10, 5, 8	0.4	0.65	25, 25, 20
SSN	0.25	0.15	45, 45, 40	0.3	0.15	45, 45, 40	0.3	0.15	45, 45, 40

(a) Configurations in use when all the objects are considered

	Mean pkl			Median pkl			Maximum pkl		
Model	θ	κ	D, R, T	θ	κ	D, R, T	θ	κ	D, R, T
FCOS3D	0.15	0.20	50, 10, 16	0.20	0.25	50, 50, 24	0.15	0.40	50, 50, 24
PGD	0.1	0.20	50, 50, 24	0.1	0.20	50, 50, 24	0.05	0.40	50, 50, 24
POINTP	0.55	0.65	5, 5, 4	0.55	0.50	5, 5, 4	0.55	0.65	15, 5, 4
REG	0.55	0.60	5, 5, 4	0.55	0.65	5, 5, 4	0.55	0.65	10, 10, 20
SECFPN	0.55	0.65	10, 5, 8	0.55	0.65	5, 10, 20	0.55	0.65	15, 30, 20
SSN	0.35	0.10	50, 50, 24	0.35	0.10	50, 50, 24	0.30	0.40	30, 10, 24

(b) Configurations in use for the cars only case

TABLE IV
PKL IMPROVEMENT WHEN MULTIPLE THRESHOLDS ARE USED.

	Model	pkl	λ	κ	θ	D, R, T
Mean pkl	FCOS3D	3.97	1.10	0.95	0.15	25, 15, 4
	PGD	4.62	1.50	0.65	0.05	15, 10, 16
	POINTP	40.14	1.35	0.95	0.70	15, 5, 4
	REG	42.56	1.30	0.95	0.75	10, 10, 4
	SECFPN	37.11	1.40	0.95	0.65	20, 10, 4
	SSN	3.25	1.10	0.30	0.30	10, 20, 4
Median pkl	FCOS3D	1.15	1.10	0.95	0.15	25, 15, 4
	PGD	1.05	1.20	0.5	0.05	20, 10, 12
	POINTP	7.83	1.35	0.95	0.70	15, 5, 4
	REG	9.35	1.30	0.95	0.75	10, 10, 4
	SECFPN	9.05	1.40	0.95	0.65	20, 10, 4
	SSN	0.60	1.10	0.90	0.30	30, 50, 8
Max pkl	FCOS3D	62.01	1.30	0.70	0.15	10, 10, 4
	PGD	116.46	1.35	0.85	0.05	10, 10, 8
	POINTP	345.00	1.4	0.8	0.65	20, 15, 10
	REG	344.41	1.35	0.8	0.7	20, 10, 4
	SECFPN	343.20	1.5	0.4	0.6	15, 15, 4
	SSN	69.92	1.25	0.8	0.2	10, 30, 8

(a) pkl values when all the objects are considered.

	Model	pkl	λ	κ	θ	D, R, T
Mean pkl	FCOS3D	1.99	1.20	0.40	0.20	5, 10, 8
	PGD	2.79	1.10	0.85	0.10	35, 45, 12
	POINTP	28.67	1.10	0.90	0.90	10, 5, 4
	REG	36.44	1.50	0.60	0.65	5, 5, 4
	SECFPN	25.56	1.60	0.90	0.75	10, 5, 4
	SSN	2.05	1.35	0.80	0.40	55, 10, 12
Median pkl	FCOS3D	0.59	1.35	0.5	0.25	55, 25, 12
	PGD	0.45	1.3	0.90	0.05	25, 5, 4
	POINTP	4.68	1.5	0.9	0.8	10, 5, 4
	REG	6.11	1.5	0.6	0.65	5, 5, 4
	SECFPN	5.80	1.6	0.9	0.75	10, 10, 12
	SSN	0.34	1.4	0.90	0.20	5, 5, 4
Max pkl	FCOS3D	32.31	0.90	0.15	5	5, 5, 12
	PGD	71.52	0.5	0.05	10	10, 10, 4
	POINTP	300.27	0.8	0.8	20	20, 5, 4
	REG	281.29	0.80	0.6	5	5, 5, 4
	SECFPN	264.57	0.90	0.60	5	5, 5, 4
	SSN	65.62	0.5	0.40	25	25, 10, 4

(b) pkl values when only cars are considered.

pkl differs the most from the ground truth. This suggests that the effectiveness of the driving task is in general improved, except for particularly challenging scenarios, where filtering objects as in the approach for *RQ1* removes too many objects, occasionally leading to a worse trajectory.

The parameters under which the new pkl value was computed are reported in Table III. Recall that those parameters are the configuration in which the best (i.e., lowest) pkl value was achieved, for each detector. As we can see from Table III, configuration parameters are strictly dependent on the algorithm. The confidence threshold is always very similar to Table II, while the criticality threshold κ and D, R, T vary significantly among the six algorithms. In general, high values of D, R, T mean that we are also including objects that are relatively distant from the ego vehicle. This happens for FCOS3D, PGD, and SSN which also have a better baseline pkl. Low values of D, R, T tend to exclude a higher number of objects. This is the case for POINTP, REG, and SECFPN. With these detectors, the trajectory is computed only relying on awareness of the vehicles that are very close to the ego vehicle. Still, the trajectory is in general improved, which suggests that those detectors provide unreliable detection results for more distant objects.

B. RQ2: Further safety and effectiveness improvement

Last, we apply the solution described in *RQ2* to improve safety and effectiveness, where we further exploit criticality information. In this case we use a combination of predicted criticality and detection confidence to decide whether to retain a predicted object (refer to Section V-B2).

The best pkl results obtained with this approach are reported in Table IV. Only the 8 underlined values resulted in a worse pkl than the approach for *RQ1*, while the 2 values in red are the only cases in which pkl is worse than the baseline pkl from Table II. In the large majority of cases (28 out of 36) results are satisfying, meaning that we obtain an improvement in the quality of the trajectory. Those results suggests that more effective ways to exploit information on object criticality exist, and further research is needed in such direction.

VII. CONCLUDING REMARKS

A. Summary

This paper advocates that the relevance of an object, as predicted by an object detector, is an important input to increase the effectiveness and safety of the driving task. The paper explores alternatives to improve trajectory planning relying on such information, showing that better (safer, and more effective) trajectories are computed. This concept builds on a paradigm shift where the quality of an object detector is not weighted based on the correctness of the detections, but on the impact of the detections on the driving task.

B. Limitations and Future Work

1) Extending the validation set: This preliminary work is limited by the reduced validation set. This was necessary for computational time, since we have explored and reported results on multiple configurations of various object detection approaches. Building on the results here discussed, it is possible to focus on fewer object detectors and specific configurations, and consequently perform a more thorough analysis.

2) Further exploitation of object criticality: Our strategy for exploiting predicted criticality, even when investigating *RQ2*, is still very simple. The relation between θ (detection confidence) and κ (object relevance) may be more complex. We need to identify a relation between detection confidence and predicted relevance, such that an improvement in both effectiveness and safety is obtained. To decide if a BB should be maintained or discarded, we plan to investigate alternative strategies, including the application of decision trees.

3) More accurate investigation of safety aspects: Evidence of safety improvement should be further detailed. The fitness of pkl as a safety metric for object detection has been analyzed experimentally in [34], where the impact of object detection errors on such metric is evaluated from different perspectives. Results showed that, although pkl is a good indicator of the quality of the planned trajectory, it is not able to differentiate between safe and hazardous deviations from the ground truth trajectory. Therefore, a different approach must be adopted to identify possible collisions and to estimate the safety of different trajectories.

4) Relevance model: Last, we are using a criticality model specifically crafted for vehicles [8] for all the objects in the dataset, while other models could be better suited for pedestrians [5], or for static objects like traffic cones. Further, a similar approach could be investigated with other definition of object relevance, and potentially for other downstream tasks.

References

[1] C. Premebida, G. Melotti, and A. Asvadi, "Rgb-d object classification for autonomous driving perception," *RGB-D Image Analysis and Processing*, pp. 377–395, 2019.

[2] S. Teng *et al.*, "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Tran. on Intelligent Vehicles*, 2023.

[3] P. S. Chib and P. Singh, "Recent advancements in end-to-end autonomous driving using deep learning: A survey," *IEEE Tran. on Intelligent Vehicles*, vol. 9, no. 1, pp. 103–118, 2024.

[4] R. McAllister, B. Wulfe, J. Mercat, L. Ellis, S. Levine, and A. Gaidon, "Control-aware prediction objectives for autonomous driving," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 01–08.

[5] M. Lyssenko, C. Gladisch, C. Heinzemann, M. Woehrle, and R. Triebel, "From evaluation to verification: Towards task-oriented relevance metrics for pedestrian detection in safety-critical domains," in *2021 IEEE/CVF CVPR Workshops (CVPRW)*, 2021, pp. 38–45.

[6] M. Wolf, L. R. Douat, and M. Erz, "Safety-aware metric for people detection," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2021-September, 2021, pp. 2759–2765.

[7] G. Volk, J. Gamerdinger, A. V. Betnuth, and O. Bringmann, "A comprehensive safety metric to evaluate perception in autonomous systems," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020*, 2020.

[8] A. Ceccarelli and L. Montecchi, "Evaluating Object (Mis)Detection From a Safety and Reliability Perspective: Discussion and Measures," *IEEE Access*, vol. 11, pp. 44 952–44 963, 5 2023.

[9] H. Caesar *et al.*, "Nuscenes: A multimodal dataset for autonomous driving," in *Proc. of the CVPR*, 2020, pp. 11 618–11 628.

[10] J. Philion, A. Kar, and S. Fidler, "Implementing planning kl-divergence," in *Computer Vision – ECCV 2020 Workshops*, A. Bartoli and A. Fusiello, Eds. Cham: Springer International Publishing, 2020, pp. 11–18.

[11] C. Sun, R. Zhang, Y. Lu, Y. Cui, Z. Deng, D. Cao, and A. Khajepour, "Toward ensuring safety for autonomous driving perception: Standardization progress, research advances, and perspectives," *IEEE Tran. on Intelligent Transportation Systems*, 2023.

[12] S. Topan *et al.*, "Interaction-dynamics-aware perception zones for obstacle detection safety evaluation," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 1201–1210.

[13] H.-C. Liao, C.-H. Cheng, H. Esen, and A. Knoll, "Improving the safety of 3d object detectors in autonomous driving using iogt and distance measures," *arXiv preprint arXiv:2209.10368*, 2022.

[14] N. Wang, Y. Luo, T. Sato, K. Xu, and Q. A. Chen, "Does physical adversarial example really matter to autonomous driving? towards system-level effect of adversarial object evasion attack," in *Proc. of ICCV*, 2023, pp. 4412–4423.

[15] T. Schreier, K. Renz, A. Geiger, and K. Chitta, "On offline evaluation of 3d object detection for autonomous driving," in *Proc. of ICCV*, 2023, pp. 4084–4089.

[16] J. Philion, A. Kar, and S. Fidler, "Learning to evaluate perception models using planner-centric metrics," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 052–14 061.

[17] L. Jiao *et al.*, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128 837–128 868, 2019.

[18] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 international conference on systems, signals and image processing (IWSSIP)*. IEEE, 2020, pp. 237–242.

[19] G. Salton and M. J. McGill, *Introduction to modern information retrieval*. mcgraw-hill, 1983.

[20] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[21] MMDetection3D Contributors, "MMDetection3D: OpenMMLab next-generation platform for general 3D object detection," https://github.com/open-mmlab/mmdetection3d (Accessed: June 7, 2023).

[22] K. Chen *et al.*, "MMDetection: Open MMLab Detection Toolbox and Benchmark," arXiv 1906.07155, 2019.

[23] *Anonymous authors*, "Github repository https://anonymous.4open.science/r/detectorAndTrajectory-D039/README.md," online, 2023.

[24] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, 2019, pp. 12 689–12 697.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[26] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, 2017, pp. 936–944.

[27] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[28] X. Zhu, Y. Ma, T. Wang, Y. Xu, J. Shi, and D. Lin, "Ssn: Shape signature networks for multi-class object detection from point clouds," in *ECCV 2020*. Springer, 2020, pp. 581–597.

[29] T. Wang, X. Zhu, J. Pang, and D. Lin, "Fcos3d: Fully convolutional one-stage monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 913–922.

[30] T. Wang, Z. Xinge, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," in *Conference on Robot Learning*. PMLR, 2022, pp. 1475–1485.

[31] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proc. of CVPR*, 2020, pp. 10 425–10 433.

[32] nuScenes Contributors, "nuScenes devkit," https://github.com/nutonomy/nuscenes-devkit (Accessed: June 7, 2023).

[33] J. Philion, A. Kar, and S. Fidler, "pkl library," https://pypi.org/project/planning-centric-metrics, 2020.

[34] A. Rønnestad, A. Ceccarelli, and L. Montecchi, "Validation of Safety Metrics for Object Detectors in Autonomous Driving," in *39th Annual ACM Symposium on Applied Computing (SAC 2024)*, Avila, Spain, 2024.