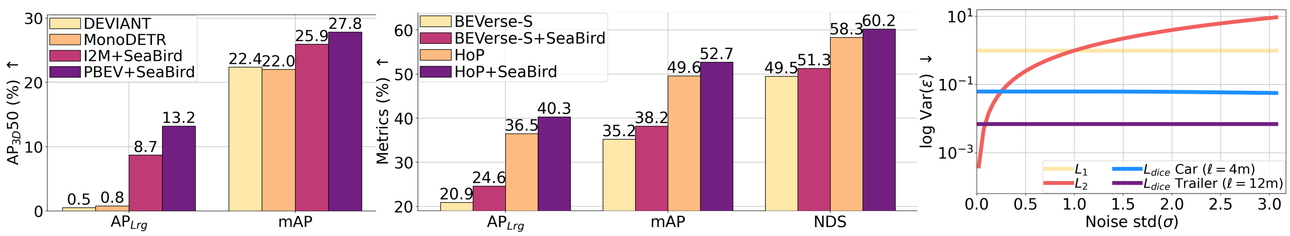
SeaBird: Segmentation in Bird's View with Dice Loss Improves Monocular 3D Detection of Large Objects

Abhinav Kumar[1] Yuliang Guo[2] Xinyu Huang[2] Liu Ren[2] Xiaoming Liu[1]
[1]Michigan State University [2]Bosch Research North America, Bosch Center for AI
[1] [kumarab6,liuxm]@msu.edu [2] [yuliang.guo2,xinyu.huang,liu.ren]@us.bosch.com
https://github.com/abhi1kumar/SeaBird



(a) Improve KITTI-360 Val SoTA. (b) Improve nuScenes Val SoTA. (c) Theory Advancement.

Figure 1. **Teaser (a)** SoTA frontal detectors struggle with large objects (low AP_{Lrg}) even on a nearly balanced KITTI-360 dataset (Skewness in Fig. 7). Our proposed SeaBird achieves significant Mono3D improvements, particularly for large objects. **(b)** SeaBird also improves two SoTA BEV detectors, BEVerse-S [116] and HoP [121] on the nuScenes dataset, particularly for large objects. **(c)** Plot of convergence variance $\text{Var}(\epsilon)$ of dice and regression losses with the noise σ in depth prediction. The y-axis denotes the deviation from the optimal weight, so the lower the better. SeaBird leverages **dice loss**, which we prove is more noise-robust than regression losses for large objects.

Abstract

Monocular 3D detectors achieve remarkable performance on cars and smaller objects. However, their performance drops on larger objects, leading to fatal accidents. Some attribute the failures to training data scarcity or the receptive field requirements of large objects. In this paper, we highlight this understudied problem of generalization to large objects. We find that modern frontal detectors struggle to generalize to large objects even on nearly balanced datasets. We argue that the cause of failure is the sensitivity of depth regression losses to noise of larger objects. To bridge this gap, we comprehensively investigate regression and dice losses, examining their robustness under varying error levels and object sizes. We mathematically prove that the dice loss leads to superior noise-robustness and model convergence for large objects compared to regression losses for a simplified case. Leveraging our theoretical insights, we propose SeaBird (Segmentation in Bird's View) as the first step towards generalizing to large objects. SeaBird effectively integrates BEV segmentation on foreground objects for 3D detection, with the segmentation head trained with the dice loss. SeaBird achieves SoTA results on the KITTI-360 leaderboard and improves existing detectors on the nuScenes leaderboard, particularly for large objects.

1. Introduction

Monocular 3D object detection (Mono3D) task aims to estimate both the 3D position and dimensions of objects in a scene from a single image. Its applications span autonomous driving [43, 50, 74], robotics [84], and augmented reality [1, 70, 76, 110], where accurate 3D understanding of the environment is crucial. Our study focuses explicitly on 3D object detectors applied to autonomous vehicles (AVs), considering the challenges and motivations deviate drastically across different applications.

AVs demand object detectors that generalize to diverse intrinsics [6], camera-rigs [35, 39], rotations [72], weather and geographical conditions [21] and also are robust to adversarial examples [120]. Since each of these poses a significant challenge, recent works focus exclusively on the generalization of object detectors to all these out-of-distribution shifts. However, our focus is on the generalization of another type, which, thus far, has been understudied in the literature – *Mono3D generalization to large objects.*

Large objects like trailers, buses and trucks are harder to detect [102] in Mono3D, sometimes resulting in fatal accidents [8, 24]. Some attribute these failures to training data scarcity [119] or the receptive field requirements [102] of large objects, but, to the best of our knowledge, no existing

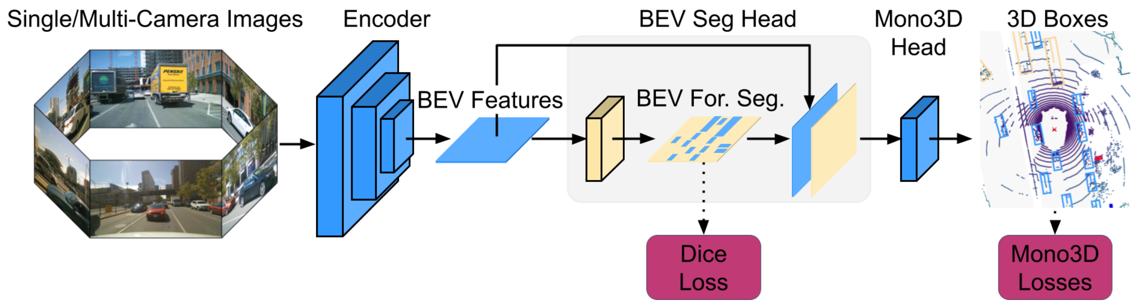


Figure 2. **SeaBird Pipeline.** SeaBird uses the predicted BEV foreground segmentation (For. Seg.) map to predict accurate 3D boxes for large objects. SeaBird training protocol involves BEV segmentation pre-training with the noise-robust dice loss and Mono3D fine-tuning.

literature provides a comprehensive analytical explanation for this phenomenon. The goal of this paper is, thus, to bring understanding and a first analytical approach to this real-world problem in the AV space – Mono3D generalization to large objects.

We conjecture that the generalization issue stems not only from limited training data or larger receptive field but also from the noise sensitivity of depth regression losses in Mono3D. To substantiate our argument, we analyze the Mono3D performance of state-of-the-art (SoTA) frontal detectors on the KITTI-360 dataset [52], which includes almost equal number ($1:2$) of large objects and cars. We observe that SoTA detectors struggle with large objects on this dataset (Fig. 1a). Next, we carefully investigate the SGD convergence of losses used in Mono3D task and mathematically prove that the dice loss, widely used in BEV segmentation, exhibits superior noise-robustness than the regression losses, particularly for large objects (Fig. 1c). Thus, the dice loss facilitates better model convergence than regression losses, improving Mono3D of large objects.

Incorporating dice loss in detection introduces unique challenges. Firstly, the dice loss does not apply to sparse detection centers and only incorporates depth information when used in the BEV space. Secondly, naive joint training of Mono3D and BEV segmentation tasks with image inputs does not always benefit Mono3D task [50, 69] due to negative transfer [19], and the underlying reasons remain unclear. Fortunately, many Mono3D segmentors and detectors are in the BEV space, where the BEV segmentor can seamlessly apply dice loss and the BEV detector can readily benefit from the segmentor in the same space. To mitigate negative transfer, we find it effective to train the BEV segmentation head on the foreground detection categories.

Building upon our theoretical findings about the dice loss, we propose a simple and effective pipeline called Segmentation in Bird's View (SeaBird) for enhancing Mono3D of large objects. SeaBird employs a sequential approach for the BEV segmentation and Mono3D heads (Fig. 2). SeaBird first utilizes a BEV segmentation head to predict the segmentation of only foreground objects, supervised by the dice loss. The dice loss offers superior noise-robustness for large objects, ensuring stable convergence, while focusing on foreground objects in segmentation mitigates negative transfer. Subsequently, SeaBird concatenates the resulting BEV segmentation map with the original BEV features as an additional feature channel and feeds this concatenated feature to a Mono3D head supervised by Mono3D losses[1]. Building upon this, we adopt a two-stage training pipeline: the first stage exclusively focuses on training the BEV segmentation head with dice loss, which fully exploits its noise-robustness and superior convergence in localizing large objects. The second stage involves both the detection loss and dice loss to finetune the Mono3D head.

In our experiments, we first comprehensively evaluate SeaBird and conduct ablations on the balanced single-camera KITTI-360 dataset [52]. SeaBird outperforms the SoTA baselines by a substantial margin. Subsequently, we integrate SeaBird as a plug-in-and-play module into two SoTA detectors on the multi-camera nuScenes dataset [7]. SeaBird again significantly improves the original detectors, particularly on large objects. Additionally, SeaBird consistently enhances Mono3D performance across backbones with those two SoTA detectors (Fig. 1b), demonstrating its utility in both edge and cloud deployments.

In summary, we make the following contributions:

- We highlight the understudied problem of generalization to large objects in Mono3D, showing that even on nearly balanced datasets, SoTA frontal models struggle to generalize due to the noise sensitivity of regression losses.
- We mathematically prove that the dice loss leads to superior noise-robustness and model convergence for large objects compared to regression losses for a simplified case and provide empirical support for more general settings.
- We propose SeaBird, which treats BEV segmentation head on foreground objects and Mono3D head sequentially and trains in a two-stage protocol to fully harness the noise-robustness of the dice loss.
- We empirically validate our theoretical findings and show significant improvements, particularly for large objects, on both KITTI-360 and nuScenes leaderboards.

[1] Only Mono3D head predicts additional 3D attributes, namely object's height and elevation.

2. Related Work

Mono3D. Mono3D popularity stems from its high accessibility from consumer vehicles compared to LiDAR/Radar-based detectors [61, 86, 109] and computational efficiency compared to stereo-based detectors [13]. Earlier approaches [12, 78] leverage hand-crafted features, while the recent ones use deep networks. Advancements include introducing new architectures [33, 88, 105], equivariance [11, 43], losses [4, 14], uncertainty [41, 63] and incorporating auxiliary tasks such as depth [71, 115], NMS [42, 56, 87], corrected extrinsics [118], CAD models [10, 45, 60] or LiDAR [81] in training. A particular line of work called Pseudo-LiDAR [65, 96] shows generalization by first estimating the depth, followed by a point cloud-based 3D detector.

Another line of work encodes image into latent BEV features [68] and attaches multiple heads for downstream tasks [116]. Some focus on pre-training [103] and rotation-equivariant convolutions [23]. Others introduce new coordinate systems [36], queries [49, 64], or positional encoding [89] in a transformer-based detection framework [9]. Some use pixel-wise depth [32], object-wise depth [16, 17, 54], or depth-aware queries [112], while many utilize temporal fusion [5, 58, 92, 101] to boost performance. A few use longer frame history [75, 121], distillation [40, 100] or stereo [47, 101]. We refer to [67, 69] for the survey. SeaBird also builds upon the BEV-based framework since it flexibly accepts single or multiple images as input and uses dice loss. Different from the majority of other detectors, SeaBird improves Mono3D of large objects using the power of dice loss. SeaBird is also the first work to mathematically prove and justify this loss choice for large objects.

BEV Segmentation. BEV segmentation typically utilizes BEV features transformed from 2D image features. Various methods encode single or multiple images into BEV features using MLPs [73] or transformers [82, 83]. Some employ learned depth distribution [30, 79], while others use attention [83, 117] or attention fields [15]. Image2Maps [83] utilizes polar ray, while PanopticBEV [27] uses transformers. FIERY [30] introduces uncertainty modelling and temporal fusion, while Simple-BEV [28] uses radar aggregation. Since BEV segmentation lacks object height and elevation, one also needs a Mono3D head to predict 3D boxes.

Joint Mono3D and BEV Segmentation. Joint 3D detection and BEV segmentation using LiDAR data [22, 86] as input benefits both tasks [95, 106]. However, joint learning on image data often hinders detection performance [50, 69, 103, 116], while the BEV segmentation improvement is inconsistent across categories [69]. Unlike these works which treat the two heads in parallel and decrease Mono3D performance [69], SeaBird treats the heads sequentially and increases Mono3D performance, particularly for large objects.

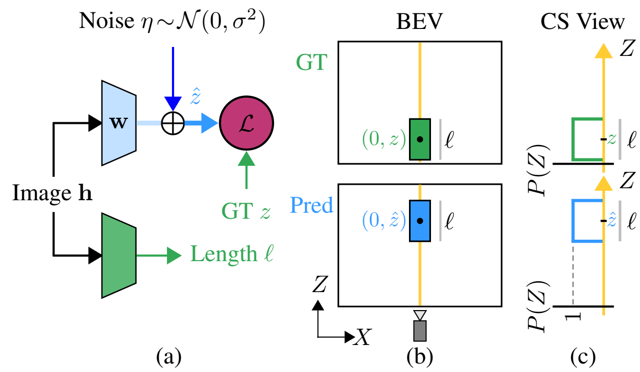


Figure 3. **(a) Problem setup**. The single-layer neural network takes an image $\mathbf{h}$ (or its features) and predicts depth $\hat{z}$ and the object length ℓ. The noise η is the additive error in depth prediction and is a normal random variable. The GT depth z supervises the predicted depth $\hat{z}$ with a loss $\mathcal{L}$ in training. We assume the network predicts the GT length ℓ. Frontal detectors directly regress the depth with $\mathcal{L}_1$, $\mathcal{L}_2$, or Smooth $\mathcal{L}_1$ loss, while SeaBird projects to BEV plane and supervises through dice loss $\mathcal{L}_{dice}$. **(b) Shifting of predictions** in BEV along the ray due to the noise η. **(c) Cross Section (CS) view** along the ray with classification scores $P(Z)$.

3. SeaBird

SeaBird is driven by a deep understanding of the distinctions between monocular regression and BEV segmentation losses. Thus, in this section, we delve into the problem and discuss existing results. We then present our theoretical findings and, subsequently, introduce our pipeline.

We introduce the problem and refer to Lemma 1 from the literature [44, 85], which *evaluates* loss quality by measuring the deviation of trained weight (after SGD updates) from the optimal weight. Fig. 3a illustrates the problem setup. Figs. 3b and 3c visualize the BEV and cross-section view, respectively. Since this deviation depends on the gradient variance of losses, we next derive the gradient variance of the dice loss in Lemma 2. By comparing the distance between trained weight and optimal weight, we assess the effectiveness of dice loss versus MAE ($\mathcal{L}_1$) and MSE ($\mathcal{L}_2$) losses in Lemma 3, and choose the representation and loss combination. Combining these findings, we establish Theorem 1 that the model trained with dice loss achieves better AP than the model trained with regression losses. Finally, we present our pipeline, SeaBird, which integrates BEV segmentation supervised by dice loss for Mono3D.

3.1. Background and Problem Statement

Mono3D networks [43, 63] commonly employ regression losses, such as $\mathcal{L}_1$ or $\mathcal{L}_2$ loss, to compare the predicted depth with ground truth (GT) depth [43, 116]. In contrast, BEV segmentation utilizes dice loss [83] or cross-entropy loss [30] at each BEV location, comparing it with GT. Despite these distinct loss functions, we evaluate their effectiveness under an idealized model, where we measure the

Table 1. **Convergence variance** of training loss functions. Gradient variance of $\mathcal{L}_{dice}$ is more noise-robust for large objects, resulting in better detectors. We do not analyze cross-entropy loss theoretically since its Var(ϵ) is infinite, but empirically in Tab. 5.

Loss $\mathcal{L}$	Gradient ϵ	Var(ϵ) (↓)
$\mathcal{L}_1$ [85] (App. A1.2.1)	$\text{sgn}(\eta)$	1
$\mathcal{L}_2$ [85] (App. A1.2.2)	η	σ^2
Dice (Lemma 2)	$\begin{cases} \frac{\text{sgn}(\eta)}{\ell} & , \lvert\eta\rvert \le \ell \\ 0 & , \lvert\eta\rvert \ge \ell \end{cases}$	$\frac{1}{\ell^2}\text{Erf}\left(\frac{\ell}{\sqrt{2}\sigma}\right)$

model *quality* by the expected deviation of trained weight (after SGD updates) from the optimal weight [85].

Lemma 1. ***Convergence analysis*** *[85]. Consider a linear regression model with trainable weight* $\mathbf{w}$ *for depth prediction* $\hat{z}$ *from an image* $\mathbf{h}$. *Assume the noise* η *is an additive error in depth prediction and is a normal random variable* $\mathcal{N}(0, \sigma^2)$. *Also, assume SGD optimizes the model parameters with loss function* $\mathcal{L}$ *during training with square summable steps* s_j, *i.e.* $s = \lim_{t\to\infty} \sum_{j=1}^{t} s_j^2$ *exists and* η *is independent of the image. Then, the expected deviation of the trained weight* ${}^{\mathcal{L}}\mathbf{w}_\infty$ *from the optimal weight* $\mathbf{w}_*$ *obeys*

$$\mathbb{E}\left(\left\|{}^{\mathcal{L}}\mathbf{w}_\infty - \mathbf{w}_*\right\|_2^2\right) = c_1 Var(\epsilon) + c_2, \tag{1}$$

where $\epsilon = \frac{\partial \mathcal{L}(\eta)}{\partial \eta}$ *is the gradient of the loss* $\mathcal{L}$ *wrt noise,* $c_1 = s\mathbb{E}(\mathbf{h}^T\mathbf{h})$ *and* c_2 *are constants independent of the loss.*

We refer to Sec. A1.1 for the proof. Eq. (1) demonstrates that training losses $\mathcal{L}$ exhibit varying gradient variances Var(ϵ). Hence, comparing this term for different losses allows us to evaluate their quality.

3.2. Loss Analysis: Dice vs. Regression

Given that [85] provides the gradient variance Var(ϵ), for $\mathcal{L}_1$ and $\mathcal{L}_2$ losses, we derive the corresponding gradient variance for dice loss in this paper to facilitate comparison. First, we express the dice loss, $\mathcal{L}_{dice}$, as a function of noise η as per its definition from [83] for Fig. 3c as:

$$\mathcal{L}_{dice}(\eta) = 1 - 2\frac{\text{Pred GT}}{\text{Pred} + \text{GT}} = \begin{cases} 1 - 2\frac{\ell - |\eta|}{2\ell} & , |\eta| \le \ell \\ 1 & , |\eta| \ge \ell \end{cases}$$

$$\implies \mathcal{L}_{dice}(\eta) = \begin{cases} \frac{|\eta|}{\ell} & , |\eta| \le \ell \\ 1 & , |\eta| \ge \ell \end{cases}, \tag{2}$$

where ℓ denotes the object length. Eq. (2) shows that the dice loss $\mathcal{L}_{dice}$ depends on the object size ℓ. With the given dice loss $\mathcal{L}_{dice}$, we proceed to derive the following lemma:

Lemma 2. ***Gradient variance of dice loss.*** *Let* $\eta = \mathcal{N}(0, \sigma^2)$ *be an additive normal random variable and* ℓ *be the object length. Let Erf be the error function. Then, the gradient variance of the dice loss* $Var_{dice}(\epsilon)$ *wrt noise* η *is*

$$Var_{dice}(\epsilon) = \frac{1}{\ell^2} Erf\left(\frac{\ell}{\sqrt{2}\sigma}\right). \tag{3}$$

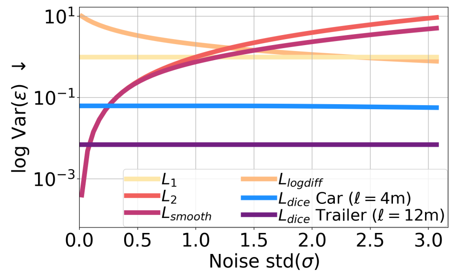


Figure 4. **Plot of convergence variance** Var(ϵ) of loss functions with the noise σ. Dice loss has minimum convergence variance with large noise, resulting in better detectors for large objects.

We refer to Sec. A1.2.3 for the proof. Eq. (3) shows that gradient variance of the dice loss $\text{Var}_{dice}(\epsilon)$ also varies inversely to the object size ℓ and the noise deviation σ (See Sec. A1.5). These two properties of dice loss are particularly beneficial for large objects.

Tab. 1 summarizes these losses, their gradients, and gradient variances. With $\text{Var}_{dice}(\epsilon)$ derived for the dice loss, we now compare the deviation of trained weight with the deviations from $\mathcal{L}_1$ or $\mathcal{L}_2$ losses, leading to our next lemma.

Lemma 3. ***Dice model is closer to optimal weight than regression loss models.*** *Based on Lemma 1 and assuming the object length* ℓ *is a constant, if* σ_m *is the solution of the equation* $\sigma^2 = \frac{1}{\ell^2} Erf\left(\frac{\ell}{\sqrt{2}\sigma}\right)$ *and the noise deviation* $\sigma \ge \sigma_c = \max\left(\sigma_m, \frac{\sqrt{2}}{\ell} Erf^{-1}(\ell^2)\right)$, *then the converged weight* ${}^{d}\mathbf{w}_\infty$ *with the dice loss* $\mathcal{L}_{dice}$ *is better than the converged weight* ${}^{r}\mathbf{w}_\infty$ *with the* $\mathcal{L}_1$ *or* $\mathcal{L}_2$ *loss, i.e.*

$$\mathbb{E}\left(\left\|{}^{d}\mathbf{w}_\infty - \mathbf{w}_*\right\|_2\right) \le \mathbb{E}\left(\left\|{}^{r}\mathbf{w}_\infty - \mathbf{w}_*\right\|_2\right). \tag{4}$$

We refer to Sec. A1.3 for the proof. Beyond noise deviation threshold $\sigma_c = \max\left(\sigma_m, \frac{\sqrt{2}}{\ell}\text{Erf}^{-1}(\ell^2)\right)$, the convergence gap between dice and regression losses widens as the object size ℓ increases. Fig. 4 depicts the superior convergence of dice loss compared to regression losses under increasing noise deviation σ pictorially. Taking the car category with $\ell = 4m$ and the trailer category with $\ell = 12m$ as examples, the noise threshold σ_c, beyond which dice loss exhibits better convergence, are $\sigma_c = 0.3m$ and $\sigma_c = 0.1m$ respectively. Combining these lemmas, we finally derive:

Theorem 1. ***Dice model has better*** AP_{3D}. *Assume the object length* ℓ *is a constant and depth is the only source of error for detection. Based on Lemma 1, if* σ_m *is the solution of the equation* $\sigma^2 = \frac{1}{\ell^2} Erf\left(\frac{\ell}{\sqrt{2}\sigma}\right)$ *and the noise deviation* $\sigma \ge \sigma_c = \max\left(\sigma_m, \frac{\sqrt{2}}{\ell} Erf^{-1}(\ell^2)\right)$, *then the Average Precision* (AP_{3D}) *of the dice model is better than* AP_{3D} *from* $\mathcal{L}_1$ *or* $\mathcal{L}_2$ *model.*

We refer to Sec. A1.4 and Tab. 8 for the proof and assumption comparisons respectively.

3.3. Discussions

Comparing classification and regression losses. We now explain how we compare classification (dice) and regression losses. Our analysis assumes one-class classification in BEV segmentation with perfect predicted foreground scores $P(Z) = 1$ (Fig. 3c). Hence, dice analysis focuses on object localization along the BEV ray (Fig. 3b) instead of classification probabilities thus allowing comparison of dice and regression losses. Lemma 1 links these losses by comparing the deviation of learned and optimal weights.

Regression losses work better than dice loss for regression tasks? Our key message is NOT always! We mathematically and empirically show that regression losses work better only when the **noise** σ **is less** in Fig. 4.

3.4. SeaBird Pipeline

Architecture. Based on theoretical insights of Theorem 1, we propose SeaBird, a novel pipeline, in Fig. 2. To effectively involve the dice loss which originally designed for segmentation task to assist Mono3D, SeaBird treats BEV segmentation of foreground objects and Mono3D head sequentially. Although BEV segmentation map provides depth information (hardest [43, 66] Mono3D parameter), it lacks elevation and height information for Mono3D task. To address this, SeaBird concatenates BEV features with predicted BEV segmentation (Fig. 2), and feeds them into the detection head to predict 3D boxes in a 7-DoF representation: BEV 2D position, elevation, 3D dimension, and yaw. Unlike most works [50, 116] that treat segmentation and detection branches in parallel, the sequential design directly utilizes refined BEV localization information to enhance Mono3D. Ablations in Sec. 4.2 validate this design choice. We defer the details of baselines to Sec. 4. Notably, our foreground BEV segmentation supervision with dice loss does not require dense BEV segmentation maps, as we efficiently prepare them from GT 3D boxes.

Training Protocol. SeaBird trains the BEV segmentation head first, employing the dice loss between the predicted and the GT BEV semantic segmentation maps, which fully utilizes the dice loss's noise-robustness and superior convergence in localizing large objects. In the second stage, we jointly fine-tune the BEV segmentation head and the Mono3D head. We validate the effectiveness of training protocol via the ablation in Sec. 4.2.

4. Experiments

Datasets. Our experiments utilize two datasets with large objects: KITTI-360 [52] and nuScenes [7] encompassing both single-camera and multi-camera configurations. We opt for KITTI-360 instead of KITTI [25] for four reasons:

Table 2. **Datasets comparison.** We use KITTI-360 and nuScenes datasets for our experiments. See Fig. 7 for the skewness.

	KITTI[25]	Waymo[90]	KITTI-360[52]	nuScenes[7]
Large objects	×	×	✓	✓
Balanced	×	×	✓	×
BEV Seg. GT	×	✓	✓	✓
#images (k)	4	52 [43]	49	168

1) KITTI-360 includes large objects, while KITTI does not; 2) KITTI-360 exhibits a balanced distribution of large objects and cars; 3) an extended version, KITTI-360 PanopticBEV [27], includes BEV segmentation GT for ablation studies, while KITTI 3D detection and the Semantic KITTI dataset [2] do not overlap in sequences; 4) KITTI-360 contains about 10× more images than KITTI. We compare these datasets in Tab. 2 and show their skewness in Fig. 7.

Data Splits. We use the following splits of the two datasets:

- *KITTI-360 Test split*: This benchmark [52] contains 300 training and 42 testing windows. These windows contain 61,056 training and 910 testing images.
- *KITTI-360 Val split*: It partitions the official train into 239 train and 61 validation windows [52]. This split contains 48,648 training and 1,294 validation images.
- *nuScenes Test split*: It has 34,149 training and 6,006 testing samples [7] from the six cameras. This split contains 204,894 training and 36,036 testing images.
- *nuScenes Val split*: It has 28,130 training and 6,019 validation samples [7] from the six cameras. This split contains 168,780 training and 36,114 validation images.

Evaluation Metrics. We use the following metrics:

- *Detection*: KITTI-360 uses the mean $\text{AP}_{3\text{D}}50$ percentage across categories to benchmark models [52]. nuScenes [7] uses the nuScenes Detection Score (NDS) as the metric. NDS is the weighted average of mean AP (mAP) and five TP metrics. We also report mAP over large categories (truck, bus, trailers and construction vehicles), cars, and small categories (pedestrians, motorcyle, bicycle, cone and barrier) as AP_{Lrg}, AP_{Car} and AP_{Sml} respectively.
- *Semantic Segmentation*: We report mean IoU over foreground and all categories at 200×200 resolution [83, 116].

KITTI-360 Baselines and SeaBird Implementation. Our evaluation on the KITTI-360 focuses on the detectors taking single-camera image as input. We evaluate SeaBird pipelines against six SoTA frontal detectors: GrooMeD-NMS [42], MonoDLE [66], GUP Net [63], DEVIANT [43], Cube R-CNN [6] and MonoDETR [114]. The choice of these models encompasses anchor [6, 42] and anchor-free methods [43, 66], CNN [63, 66], group CNN [43] and transformer-based [114] architectures. Further, MonoDLE normalizes loss with GT box dimensions.

Due to SeaBird's BEV-based approach, we do not integrate it with these frontal view detectors. Instead, we extend two SoTA image-to-BEV segmentation methods, Image2Maps (I2M) [83] and PanopticBEV (PBEV) [27] with

Table 3. **KITTI-360 Test detection results.** SeaBird pipelines outperform all monocular baselines, and also outperform old LiDAR baselines. Click for the KITTI-360 leaderboard as well as our PBEV+SeaBird and I2M+SeaBird entries. [Key: **Best**, **Second Best**, L= LiDAR, C= Camera, †= Retrained].

Modality L	Modality C	Method	Venue	AP_{3D} 50 (↑) mAP [%]	AP_{3D} 25 (↑) mAP [%]
✓		L-VoteNet [80]	ICCV19	3.40	30.61
✓		L-BoxNet [80]	ICCV19	4.08	23.59
	✓	GrooMeD† [42]	CVPR21	0.17	16.12
	✓	MonoDLE† [66]	CVPR21	0.85	28.99
	✓	GUP Net† [63]	ICCV21	0.87	27.25
	✓	DEVIANT† [43]	ECCV22	0.88	26.96
	✓	Cube R-CNN† [6]	CVPR23	0.80	15.57
	✓	MonoDETR† [114]	ICCV23	0.79	27.13
	✓	**I2M+SeaBird**	CVPR24	**3.14**	**35.04**
	✓	**PBEV+SeaBird**	CVPR24	**4.64**	**37.12**

SeaBird. Since both BEV segmentors already include their own implementations of the image encoder, the image-to-BEV transform, and the segmentation head, implementing the SeaBird pipeline only involves adding a detection head, which we chose to be Box Net [108]. SeaBird extensions employ dice loss for BEV segmentation, Smooth $\mathcal{L}_1$ losses [26] in the BEV space to supervise the BEV 2D position, elevation, and 3D dimension, and cross entropy loss to supervise orientation.

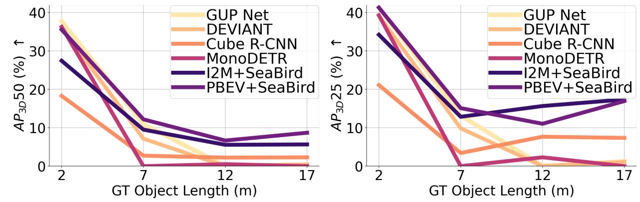
nuScenes Baselines and SeaBird Implementation. We integrate SeaBird into two prototypical BEV-based detectors, BEVerse [116] and HoP [121] to prove the effectiveness of SeaBird. Our choice of these models encompasses both transformer and convolutional backbones, multi-head and single-head architectures, shorter and longer frame history, and non-query and query-based detectors. This comprehensively allows us to assess SeaBird's impact on large object detection. BEVerse employs a multi-head architecture with a transformer backbone and shorter frame history. HoP is single-head query-based SoTA model utilizing BEVDet4D [31] with CNN backbone, and longer frame history.

BEVerse [116] includes its own implementation of detection head and BEV segmentation head in parallel. We reorganize the two heads to follow our sequential design and adhere to our training protocol for network training. Since HoP [121] lacks a BEV segmentation head, we incorporate the one from BEVerse into this HoP extension with SeaBird.

4.1. KITTI-360 Mono3D

KITTI-360 Test. Tab. 3 presents KITTI-360 leaderboard results, demonstrating the superior performance of both SeaBird pipelines compared to all monocular baselines across all metrics. Moreover, PBEV+SeaBird also outperforms both legacy LiDAR baselines on all metrics, while I2M+SeaBird surpasses them on the AP_{3D} 25 metric.

KITTI-360 Val. Tab. 4 presents the results on KITTI-360



(a) AP_{3D} 50 comparison. (b) AP_{3D} 25 comparison.

Figure 5. **Lengthwise AP Analysis** of four SoTA detectors of Tab. 4 and two SeaBird pipelines on KITTI-360 Val split. SeaBird pipelines outperform all baselines on large objects with over 10m in length.

Val split, reporting the **median** model over three different seeds with the model being the final checkpoint as [43]. SeaBird pipelines outperform all monocular baselines on all but one metric, similar to Tab. 3 results. Due to the dice loss in SeaBird, the biggest improvement shows up on larger objects. Tab. 4 also includes the upper-bound oracle, where we train the Box Net with the GT BEV segmentation maps.

Lengthwise AP Analysis. Theorem 1 states that training a model with dice loss should lead to lower errors and, consequently, a better detector for large objects. To validate this claim, we analyze the detection performance with AP_{3D} 50 and AP_{3D} 25 metrics against the object's lengths. For this analysis, we divide objects into four bins based on their GT object length (max of sizes): $[0, 5), [5, 10), [10, 15), [15+$ m. Fig. 5 shows that SeaBird pipelines excel for large objects, where the baselines' performance drops significantly.

BEV Semantic Segmentation. Tab. 4 also presents the BEV semantic segmentation results on the KITTI-360 Val split. SeaBird pipelines outperforms the baseline I2M [83], and achieve similar performance to PBEV [27] in BEV segmentation. We retrain all BEV segmentation models only on foreground detection categories for a fair comparison.

4.2. Ablation Studies on KITTI-360 Val

Tab. 5 ablates I2M [83] +SeaBird on the KITTI-360 Val split, following the experimental settings of Sec. 4.1.

Dice Loss. Tab. 5 shows that both dice loss and BEV representation are crucial to Mono3D of large objects. Replacing dice loss with MSE or Smooth $\mathcal{L}_1$ loss, or only BEV representation (w/o dice) reduces Mono3D performance.

Mono3D and BEV Segmentation. Tab. 5 shows that removing the segmentation head hinders Mono3D performance. Conversely, removing detection head also diminishes the BEV segmentation performance for the segmentation model. This confirms the mutual benefit of sequential BEV segmentation on foreground objects and Mono3D.

Semantic Category in BEV Segmentation. We next analyze whether background categories play any role in Mono3D. Tab. 5 shows that changing the foreground (For.) categories to foreground + background (All) does not help Mono3D. This aligns with the observations of [69, 103, 116] that report lower performance on joint Mono3D and

Table 4. **KITTI-360 Val detection and segmentation results.** SeaBird pipelines outperform all frontal monocular baselines, particularly for large objects. Dice loss in SeaBird also improves the BEV only (w/o dice) version of SeaBird pipelines. I2M and PBEV are BEV segmentors. So, we do not report their Mono3D performance. [Key: **Best**, **Second Best**, †= Retrained]

View	Method	BEV Seg Loss	Venue	AP_{3D} 50 [%](↑)			AP_{3D} 25 [%](↑)			BEV Seg IoU [%](↑)		
				AP_{Lrg}	AP_{Car}	mAP	AP_{Lrg}	AP_{Car}	mAP	Large	Car	M_{For}
Frontal	GrooMeD-NMS† [42]	–	CVPR21	0.00	33.04	16.52	0.00	38.21	19.11	–	–	–
	MonoDLE† [66]		CVPR21	0.94	44.81	22.88	4.64	50.52	27.58	–	–	–
	GUP Net† [63]		ICCV21	0.54	45.11	22.83	0.98	50.52	25.75	–	–	–
	DEVIANT† [43]		ECCV22	0.53	44.25	22.39	1.01	48.57	24.79	–	–	–
	Cube R-CNN† [6]		CVPR23	0.75	22.52	11.63	5.55	27.12	16.34	–	–	–
	MonoDETR† [114]		ICCV23	0.81	43.24	22.02	4.50	48.69	26.60	–	–	–
BEV	I2M† [83]	Dice	ICRA22	–	–	–	–	–	–	20.46	38.04	29.25
	I2M+SeaBird	×	CVPR24	4.86	45.09	24.98	26.33	52.31	39.32	0.00	7.07	3.54
	I2M+SeaBird	Dice	CVPR24	8.71	43.19	25.95	35.76	52.22	43.99	23.23	39.61	31.42
	PBEV† [27]	CE	RAL22	–	–	–	–	–	–	23.83	48.54	36.18
	PBEV+SeaBird	×	CVPR24	7.64	45.37	26.51	29.72	53.86	41.79	2.07	1.47	1.57
	PBEV+SeaBird	Dice	CVPR24	13.22	42.46	27.84	37.15	52.53	44.84	24.30	48.04	36.17
	Oracle (GT BEV)		–	26.77	51.79	39.28	49.74	56.62	53.18	100.00	100.00	100.00

Table 5. **Ablation studies** on KITTI-360 Val. [Key: **Best**, **Second Best**]

Changed	From → To	AP_{3D} 50 [%](↑)			AP_{3D} 25 [%](↑)			BEV Seg IoU [%](↑)			
		AP_{Lrg}	AP_{Car}	mAP	AP_{Lrg}	AP_{Car}	mAP	Large	Car	M_{For}	M_{All}
Segmentation Loss	Dice → No Loss	4.86	45.09	24.98	26.33	52.31	39.32	0.00	7.07	3.54	–
	Dice → Smooth $\mathcal{L}_1$	7.63	36.69	22.16	31.01	47.51	39.26	17.16	34.67	25.92	–
	Dice → MSE	7.04	35.59	21.32	30.90	44.71	37.81	17.46	34.85	26.16	–
	Dice → CE	7.06	35.60	21.33	33.22	47.60	40.41	21.83	38.11	29.97	–
Segmentation Head	Yes → No	7.52	39.24	23.38	31.83	47.88	39.86	–	–	–	–
Detection Head	Yes → No	–	–	–	–	–	–	20.46	38.04	29.25	–
Semantic Category	For. → All	1.61	44.12	22.87	15.36	51.76	33.56	19.26	34.46	26.86	24.34
	For. → Car	4.17	43.01	23.59	22.68	51.58	37.13	–	40.28	20.14	–
Multi-head Arch.	Sequential → Parallel	9.12	40.27	24.69	32.45	51.55	42.00	22.19	40.37	31.28	–
BEV Shortcut	Yes → No	6.53	38.12	22.33	32.05	52.62	42.34	23.00	40.39	31.70	–
Training Protocol	S+J → J [116]	7.42	42.73	25.08	31.94	49.88	40.91	22.91	39.66	31.29	–
	S+J → D+J [106]	6.07	43.43	24.75	29.24	52.96	41.10	20.71	35.68	28.20	–
I2M+SeaBird	–	8.71	43.19	25.95	35.76	52.22	43.99	23.23	39.61	31.42	–

BEV segmentation with all categories. We believe this decrease happens because the network gets distracted while getting the background right. We also predict one foreground category (Car) instead of all in BEV segmentation. Tab. 5 shows that predicting all foreground categories in BEV segmentation is crucial for overall good Mono3D.

Multi-head Architecture. SeaBird employs a sequential architecture (Arch.) of segmentation and detection heads instead of parallel architecture. Tab. 5 shows that the sequential architecture outperforms the parallel one. We attribute this Mono3D boost to the explicit object localization provided by segmentation in the BEV plane.

BEV Shortcut. Sec. 3.4 mentions that SeaBird's Mono3D head utilizes both the BEV segmentation map and BEV features. Tab. 5 demonstrates that providing BEV features to the detection head is crucial for good Mono3D. This is because the BEV map lacks elevation information, and incorporating BEV features helps estimate elevation.

Training Protocol. SeaBird trains segmentor first and then jointly trains detector and segmentor (S+J). We compare with direct joint training (J) of [116] and training detection followed by joint training (D+J) of [106]. Tab. 5 shows that SeaBird training protocol works best.

4.3. nuScenes Mono3D

We next benchmark SeaBird on nuScenes [7], which encompasses more diverse object categories such as trailers, buses, cars and traffic cones, compared to KITTI-360 [52].

nuScenes Test. Tab. 6 presents the results of incorportaing SeaBird to the HoP models with the V2-99 and R101 backbones. SeaBird with both V2-99 and R101 backbones outperform several SoTA methods on the nuScenes leaderboard, as well as the baseline HoP, on nearly every metric. Interestingly, SeaBird pipelines also outperform several baselines which use higher resolution (900×1600) inputs. Most importantly, SeaBird pipelines achieve the highest AP_{Lrg} performance, providing empirical support for the claims of Theorem 1.

nuScenes Val. Tab. 7 showcases the results of integrating SeaBird with BEVerse [116] and HoP [121] at multiple res-

Table 6. **nuScenes Test detection results.** SeaBird pipelines achieve the best AP_{Lrg} among methods without Class Balanced Guided Sampling (CBGS) [119] and future frames. Results are from the nuScenes leaderboard or corresponding papers on V2-99 or R101 backbones. [Key: **Best**, **Second Best**, S= Small, *= Reimplementation, §= CBGS, ○●= Future Frames.]

Resolution	Method	BBone	Venue	AP_{Lrg} (↑)	AP_{Car} (↑)	AP_{Sml} (↑)	mAP (↑)	NDS (↑)
512×1408	BEVDepth [48] in [37]	R101	AAAI23	–	–	–	39.6	48.3
	BEVStereo [47] in [37]	R101	AAAI23	–	–	–	40.4	50.2
	P2D [37]	R101	ICCV23	–	–	–	43.6	53.0
	BEVerse-S [116]	Swin-S	ArXiv	24.4	60.4	47.0	39.3	53.1
	HoP* [121]	R101	ICCV23	**36.0**	**65.0**	**53.9**	**47.9**	**57.5**
	HoP+SeaBird	R101	CVPR24	**36.6**	**65.8**	**54.7**	**48.6**	**57.0**
640×1600	SpatialDETR [20]	V2-99	ECCV22	30.2	61.0	48.5	42.5	48.7
	3DPPE [89]	V2-99	ICCV23	–	–	–	46.0	51.4
	$X3KD_{all}$ [40]	R101	CVPR23	–	–	–	45.6	56.1
	PETRv2 [58]	V2-99	ICCV23	36.4	66.7	55.6	49.0	58.2
	VEDet [11]	V2-99	CVPR23	**37.1**	68.5	**57.7**	50.5	58.5
	FrustumFormer [98]	V2-99	CVPR23	–	–	–	**51.6**	58.9
	MV2D [99]	V2-99	ICCV23	–	–	–	**51.1**	**59.6**
	HoP* [121]	V2-99	ICCV23	**37.1**	**68.7**	55.6	49.4	58.9
	HoP+SeaBird	V2-99	CVPR24	**38.4**	**70.2**	**57.4**	**51.1**	**59.7**
	SA-BEV§ [113]	V2-99	ICCV23	40.5	68.9	60.5	53.3	62.4
	FB-BEV§ [51]	V2-99	ICCV23	39.3	71.7	61.6	53.7	62.4
	CAPE§ [104]	V2-99	CVPR23	41.3	71.4	63.3	55.3	62.8
	SparseBEV○● [55]	V2-99	ICCV23	45.6	76.3	68.8	60.3	67.5
900×1600	ParametricBEV [107]	R101	ICCV23	–	–	–	46.8	49.5
	UVTR [46]	R101	NeurIPS22	35.1	67.3	52.9	47.2	55.1
	BEVFormer [50]	V2-99	ECCV22	34.4	67.7	55.2	48.9	56.9
	PolarFormer [36]	V2-99	AAAI23	36.8	68.4	55.5	49.3	57.2
	STXD [34]	V2-99	NeurIPS23	–	–	–	49.7	58.3

Table 7. **nuScenes Val detection results.** SeaBird pipelines outperform the two baselines BEVerse and HoP, particularly for large objects. We train all models without CBGS. See Tab. 16 for a detailed comparison. [Key: S= Small, T= Tiny, ♙= Released, *= Reimplementation]

Resolution	Method	BBone	Venue	AP_{Lrg} (↑)	AP_{Car} (↑)	AP_{Sml} (↑)	mAP (↑)	NDS (↑)
256×704	BEVerse-T♙ [116]	Swin-T	ArXiv	18.5	53.4	38.8	32.1	46.6
	+SeaBird		CVPR24	19.5 (+1.0)	54.2 (+0.8)	41.1 (+2.3)	33.8 (+1.5)	48.1 (+1.7)
	HoP♙ [121]	R50	ICCV23	27.4	57.2	46.4	39.9	50.9
	+SeaBird		CVPR24	28.2 (+0.8)	58.6 (+1.4)	47.8 (+1.4)	41.1 (+1.2)	51.5 (+0.6)
512×1408	BEVerse-S♙ [116]	Swin-S	ArXiv	20.9	56.2	42.2	35.2	49.5
	+SeaBird		CVPR24	24.6 (+3.7)	58.7 (+2.5)	45.0 (+2.8)	38.2 (+3.0)	51.3 (+1.8)
	HoP* [121]	R101	ICCV23	31.4	63.7	52.5	45.2	55.0
	+SeaBird		CVPR24	32.9 (+1.5)	65.0 (+1.3)	53.1 (+0.6)	46.2 (+1.0)	54.7 (–0.3)
640×1600	HoP* [121]	V2-99	ICCV23	36.5	69.1	56.1	49.6	58.3
	+SeaBird		CVPR24	40.3 (+3.8)	71.7 (+2.6)	58.8 (+2.7)	52.7 (+3.1)	60.2 (+1.9)

olutions, as described in [116, 121]. Tab. 7 demonstrates that integrating SeaBird consistently improves these detectors on almost every metric at multiple resolutions. The improvements on AP_{Lrg} empirically support the claims of Theorem 1 and validate the effectiveness of dice loss and BEV segmentation in localizing large objects.

5. Conclusions

This paper highlights the understudied problem of Mono3D generalization to large objects. Our findings reveal that modern frontal detectors struggle to generalize to large objects even when trained on balanced datasets. To bridge this gap, we investigate the regression and dice losses, examining their robustness under varying error levels and object sizes. We mathematically prove that the dice loss outperforms regression losses in noise-robustness and model convergence for large objects for a simplified case. Leveraging our theoretical insights, we propose SeaBird (Segmentation in Bird's View) as the first step towards generalizing to large objects. SeaBird effectively integrates BEV segmentation with the dice loss for Mono3D. SeaBird achieves SoTA results on the KITTI-360 leaderboard and consistently improves existing detectors on the nuScenes leaderboard, particularly for large objects. We hope that this initial step towards generalization will contribute to safer AVs.

References

[1] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *IJCV*, 2018. 1

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *ICCV*, 2019. 5

[3] Zygmunt Birnbaum. An inequality for Mill's ratio. *The Annals of Mathematical Statistics*, 1942. 15

[4] Garrick Brazil and Xiaoming Liu. M3D-RPN: Monocular 3D region proposal network for object detection. In *ICCV*, 2019. 3

[5] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3D object detection in monocular video. In *ECCV*, 2020. 3

[6] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3D: A large benchmark and model for 3D object detection in the wild. In *CVPR*, 2023. 1, 5, 6, 7

[7] Holger Caesar, Varun Bankiti, Alex Lang, Sourabh Vora, Venice Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2, 5, 7, 17

[8] Brittany Caldwell. 2 die when tesla crashes into parked tractor-trailer in florida. https://www.wftv.com/news/local/2-die-when-tesla-crashes-into-parked-tractor-trailer-florida/KJGMHHYTQZA2HNAHWL2OFSVIPM/, 2022. Accessed: 2023-11-06. 1

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[10] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teuliere, and Thierry Chateau. Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image. In *CVPR*, 2017. 3

[11] Dian Chen, Jie Li, Vitor Guizilini, Rares Andrei Ambrus, and Adrien Gaidon. Viewpoint equivariance for multi-view 3D object detection. In *CVPR*, 2023. 3, 8

[12] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3D object detection for autonomous driving. In *CVPR*, 2016. 3

[13] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. DSGN: Deep stereo geometry network for 3D object detection. In *CVPR*, 2020. 3

[14] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. MonoPair: Monocular 3D object detection using pairwise spatial relationships. In *CVPR*, 2020. 3

[15] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. NEAT: Neural attention fields for end-to-end autonomous driving. In *ICCV*, 2021. 3

[16] Wonhyeok Choi, Mingyu Shin, and Sunghoon Im. Depth-discriminative metric learning for monocular 3D object detection. In *NeurIPS*, 2023. 3

[17] Xiaomeng Chu, Jiajun Deng, Yuan Zhao, Jianmin Ji, Yu Zhang, Houqiang Li, and Yanyong Zhang. OA-BEV: Bringing object awareness to bird's-eye-view representation for multi-camera 3D object detection. *arXiv preprint arXiv:2301.05711*, 2023. 3

[18] MMDetection3D Contributors. MMDetection3D: Open-MMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 17

[19] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020. 2

[20] Simon Doll, Richard Schulz, Lukas Schneider, Viviane Benzin, Markus Enzweiler, and Hendrik Lensch. SpatialDETR: Robust scalable transformer-based 3D object detection from multi-view camera images with global cross-sensor attention. In *ECCV*, 2022. 8

[21] Yinpeng Dong, Caixin Kang, Jinlai Zhang, Zijian Zhu, Yikai Wang, Xiao Yang, Hang Su, Xingxing Wei, and Jun Zhu. Benchmarking robustness of 3D object detection to common corruptions. In *CVPR*, 2023. 1

[22] Lue Fan, Feng Wang, Naiyan Wang, and Zhao Zhang. Fully sparse 3D object detection. In *NeurIPS*, 2022. 3

[23] Chengjian Feng, Zequn Jie, Yujie Zhong, Xiangxiang Chu, and Lin Ma. AEDet: Azimuth-invariant multi-view 3D object detection. *arXiv preprint arXiv:2211.12501*, 2022. 3

[24] Roshan Fernandez. A tesla driver was killed after smashing into a firetruck on a california highway. https://www.npr.org/2023/02/20/1158367204/tesla-driver-killed-california-firetruck-nhtsa, 2023. Accessed: 2023-11-06. 1

[25] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 5, 16

[26] Ross Girshick. Fast R-CNN. In *ICCV*, 2015. 6

[27] Nikhil Gosala and Abhinav Valada. Bird's-eye-view panoptic segmentation using monocular frontal view images. *RAL*, 2022. 3, 5, 6, 7, 17, 18, 19

[28] Adam Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-BEV: What really matters for multi-sensor BEV perception? In *CoRL*, 2022. 3

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 18

[30] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. FIERY: future instance prediction in bird's-eye view from surround monocular cameras. In *ICCV*, 2021. 3

[31] Junjie Huang and Guan Huang. BEVDet4D: Exploit temporal cues in multi-camera 3D object detection. *arXiv preprint arXiv:2203.17054*, 2022. 6, 21

[32] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. BEVDet: High-performance multi-camera 3D object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 3, 21

[33] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston Hsu. MonoDTR: Monocular 3D object detection with depth-aware transformer. In *CVPR*, 2022. 3

[34] Sujin Jang, Dae Ung Jo, Sung Ju Hwang, Dongwook Lee, and Daehyun Ji. STXD: Structural and temporal cross-modal distillation for multi-view 3D object detection. In *NeurIPS*, 2023. 8

[35] Jinrang Jia, Zhenjia Li, and Yifeng Shi. MonoUNI: A unified vehicle and infrastructure-side monocular 3D object detection network with sufficient depth clues. In *NeurIPS*, 2023. 1

[36] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3D object detection with polar transformers. In *AAAI*, 2023. 3, 8, 21

[37] Sanmin Kim, Youngseok Kim, In-Jae Lee, and Dongsuk Kum. Predict to Detect: Prediction-guided 3D object detection using sequential images. In *ICCV*, 2023. 8, 21

[38] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 18

[39] Tzofi Klinghoffer, Jonah Philion, Wenzheng Chen, Or Litany, Zan Gojcic, Jungseock Joo, Ramesh Raskar, Sanja Fidler, and Jose Alvarez. Towards viewpoint robustness in Bird's Eye View segmentation. In *ICCV*, 2023. 1

[40] Marvin Klingner, Shubhankar Borse, Varun Ravi Kumar, Behnaz Rezaei, Venkatraman Narayanan, Senthil Yogamani, and Fatih Porikli. X3KD: Knowledge distillation across modalities, tasks and stages for multi-camera 3D object detection. In *CVPR*, 2023. 3, 8

[41] Abhinav Kumar, Tim Marks, Wenxuan Mou, Ye Wang, Michael Jones, Anoop Cherian, Toshiaki Koike-Akino, Xiaoming Liu, and Chen Feng. LUVLi face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In *CVPR*, 2020. 3

[42] Abhinav Kumar, Garrick Brazil, and Xiaoming Liu. GrooMeD-NMS: Grouped mathematically differentiable NMS for monocular 3D object detection. In *CVPR*, 2021. 3, 5, 6, 7, 18

[43] Abhinav Kumar, Garrick Brazil, Enrique Corona, Armin Parchami, and Xiaoming Liu. DEVIANT: Depth Equivariant Network for monocular 3D object detection. In *ECCV*, 2022. 1, 3, 5, 6, 7, 16, 19

[44] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an $\mathcal{O}(1/t)$ convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*, 2012. 3, 13, 14

[45] Hyo-Jun Lee, Hanul Kim, Su-Min Choi, Seong-Gyun Jeong, and Yeong Koh. BAAM: Monocular 3D pose and shape reconstruction with bi-contextual attention module and attention-guided modeling. In *CVPR*, 2023. 3

[46] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3D object detection. In *NeurIPS*, 2022. 8

[47] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. BEVStereo: Enhancing depth estimation in multi-view 3D object detection with dynamic temporal stereo. In *AAAI*, 2023. 3, 8

[48] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. BEVDepth: Acquisition of reliable depth for multi-view 3D object detection. In *AAAI*, 2023. 8, 21

[49] Yangguang Li, Bin Huang, Zeren Chen, Yufeng Cui, Feng Liang, Mingzhu Shen, Fenggang Liu, Enze Xie, Lu Sheng, Wanli Ouyang, and Jing Shao. Fast-BEV: A fast and strong bird's-eye view perception baseline. In *NeurIPS Workshops*, 2023. 3

[50] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 1, 2, 3, 5, 8, 17, 21

[51] Zhiqi Li, Zhiding Yu, Wenhai Wang, Anima Anandkumar, Tong Lu, and Jose Alvarez. FB-BEV: BEV representation from forward-backward view transformations. In *ICCV*, 2023. 8

[52] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *TPAMI*, 2022. 2, 5, 7, 17

[53] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 17

[54] Feng Liu and Xiaoming Liu. Voxel-based 3D detection and reconstruction of multiple objects from a single image. In *NeurIPS*, 2021. 3

[55] Haisong Liu Liu, Yao Teng Teng, Tao Lu, Haiguang Wang, and Limin Wang. SparseBEV: High-performance sparse 3D object detection from multi-camera videos. In *ICCV*, 2023. 8, 19

[56] Xianpeng Liu, Ce Zheng, Kelvin Cheng, Nan Xue, Guo-Jun Qi, and Tianfu Wu. Monocular 3D object detection with bounding box denoising in 3D by perceiver. In *ICCV*, 2023. 3

[57] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. PETR: Position embedding transformation for multi-view 3D object detection. In *ECCV*, 2022. 21

[58] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. PETRv2: A unified framework for 3D perception from multi-camera images. In *ICCV*, 2023. 3, 8, 21

[59] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 18

[60] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang. AutoShape: Real-time shape-aware monocular 3D object detection. In *ICCV*, 2021. 3

[61] Yunfei Long, Abhinav Kumar, Daniel Morris, Xiaoming Liu, Marcos Castro, and Punarjay Chakravarty. RADIANT: RADar Image Association Network for 3D object detection. In *AAAI*, 2023. 3

[62] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 18

[63] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncer-

tainty projection network for monocular 3D object detection. In *ICCV*, 2021. 3, 5, 6, 7, 16, 19, 20

[64] Zhipeng Luo, Changqing Zhou, Gongjie Zhang, and Shijian Lu. DETR4D: Direct multi-view 3D object detection with sparse attention. *arXiv preprint arXiv:2212.07849*, 2022. 3

[65] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving. In *ICCV*, 2019. 3

[66] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3D object detection. In *CVPR*, 2021. 5, 6, 7, 18, 19

[67] Xinzhu Ma, Wanli Ouyang, Andrea Simonelli, and Elisa Ricci. 3D object detection from images for autonomous driving: A survey. *TPAMI*, 2023. 3

[68] Xinzhu Ma, Yongtao Wang, Yinmin Zhang, Zhiyi Xia, Yuan Meng, Zhihui Wang, Haojie Li, and Wanli Ouyang. Towards fair and comprehensive comparisons for image-based 3D object detection. In *ICCV*, 2023. 3

[69] Yuexin Ma, Tai Wang, Xuyang Bai, Huitong Yang, Yuenan Hou, Yaming Wang, Yu Qiao, Ruigang Yang, Dinesh Manocha, and Xinge Zhu. Vision-centric BEV perception: A survey. *arXiv preprint arXiv:2208.02797*, 2022. 2, 3, 6

[70] Nathaniel Merrill, Yuliang Guo, Xingxing Zuo, Xinyu Huang, Stefan Leutenegger, Xi Peng, Liu Ren, and Guoquan Huang. Symmetry and uncertainty-aware object SLAM for 6DoF object pose estimation. In *CVPR*, 2022. 1

[71] Zhixiang Min, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Enrique Dunn, and Manmohan Chandraker. NeurOCS: Neural NOCS supervision for monocular 3D object localization. In *CVPR*, 2023. 3

[72] SungHo Moon, JinWoo Bae, and SungHoon Im. Rotation matters: Generalized monocular 3D object detection for various camera systems. *arXiv preprint arXiv:2310.05366*, 2023. 1

[73] Bowen Pan, Jiankai Sun, Ho Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *RAL*, 2020. 3

[74] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is Pseudo-LiDAR needed for monocular 3D object detection? In *ICCV*, 2021. 1, 18

[75] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3D object detection. In *ICLR*, 2023. 3, 21

[76] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation. In *ICCV*, 2019. 1

[77] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 17

[78] Nadia Payet and Sinisa Todorovic. From contours to 3D object detection and pose estimation. In *ICCV*, 2011. 3

[79] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In *ECCV*, 2020. 3

[80] Charles Qi, Or Litany, Kaiming He, and Leonidas Guibas. Deep hough voting for 3D object detection in point clouds. In *ICCV*, 2019. 6

[81] Cody Reading, Ali Harakeh, Julia Chae, and Steven Waslander. Categorical depth distribution network for monocular 3D object detection. In *CVPR*, 2021. 3

[82] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *CVPR*, 2020. 3

[83] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Translating images into maps. In *ICRA*, 2022. 3, 4, 5, 6, 7, 17, 18, 19

[84] Ashutosh Saxena, Justin Driemeyer, and Andrew Ng. Robotic grasping of novel objects using vision. *IJRR*, 2008. 1

[85] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *ICML*, 2007. 3, 4, 13, 14

[86] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D object proposal generation and detection from point cloud. In *CVPR*, 2019. 3

[87] Xuepeng Shi, Zhixiang Chen, and Tae-Kyun Kim. Distance-normalized unified representation for monocular 3D object detection. In *ECCV*, 2020. 3

[88] Xuepeng Shi, Zhixiang Chen, and Tae-Kyun Kim. Multivariate probabilistic monocular 3D object detection. In *WACV*, 2023. 3

[89] Changyong Shu, Fisher Yu, and Yifan Liu. 3DPPE: 3D point positional encoding for multi-camera 3D object detection transformers. In *ICCV*, 2023. 3, 8, 21

[90] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 5

[91] Mingxing Tan, Ruoming Pang, and Quoc Le. EfficientDet: Scalable and efficient object detection. In *CVPR*, 2020. 18

[92] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. StreamPETR: Exploring object-centric temporal modeling for efficient multi-view 3D object detection. In *ICCV*, 2023. 3

[93] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully convolutional one-stage monocular 3D object detection. In *ICCV Workshops*, 2021. 21

[94] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *CoRL*, 2021. 21

[95] Xueqing Wang, Diankun Zhang, Haoyu Niu, and Xiaojun Liu. Segmentation can aid detection: Segmentation-guided

single stage detection for 3D point cloud. *Electronics*, 2023. 3

[96] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Weinberger. Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving. In *CVPR*, 2019. 3

[97] Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. DETR3D: 3D object detection from multi-view images via 3D-to-2D queries. In *CoRL*, 2021. 21

[98] Yuqi Wang, Yuntao Chen, and Zhaoxiang Zhang. FrustumFormer: Adaptive instance-aware resampling for multi-view 3D detection. In *CVPR*, 2023. 8

[99] Zitian Wang, Zehao Huang, Jiahui Fu, Naiyan Wang, and Si Liu. Object as Query: Lifting any 2D object detector to 3D detection. In *ICCV*, 2023. 8

[100] Zeyu Wang, Dingwen Li, Chenxu Luo, Cihang Xie, and Xiaodong Yang. DistillBEV: Boosting multi-camera 3D object detection with cross-modal knowledge distillation. In *ICCV*, 2023. 3

[101] Zengran Wang, Chen Min, Zheng Ge, Yinhao Li, Zeming Li, Hongyu Yang, and Di Huang. STS: Surround-view temporal stereo for multi-view 3D detection. In *AAAI*, 2023. 3, 21

[102] Chen Wu. Waymo keynote talk, CVPR workshop on autonomous driving at 17:20. `https://www.youtube.com/watch?v=fXsbI2VkHgc`, 2023. Accessed: 2023-11-11. 1

[103] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose Alvarez. M^{2}BEV: Multi-camera joint 3D detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. 3, 6

[104] Kaixin Xiong, Shi Gong, Xiaoqing Ye, Xiao Tan, Ji Wan, Errui Ding, Jingdong Wang, and Xiang Bai. CAPE: Camera view position embedding for multi-view 3D object detection. In *CVPR*, 2023. 8, 21

[105] Junkai Xu, Liang Peng, Haoran Cheng, Hao Li, Wei Qian, Ke Li, Wenxiao Wang, and Deng Cai. MonoNeRD: NeRF-like representations for monocular 3D object detection. In *ICCV*, 2023. 3

[106] Haitao Yang, Zaiwei Zhang, Xiangru Huang, Min Bai, Chen Song, Bo Sun, Li Erran Li, and Qixing Huang. LiDAR-based 3D object detection via hybrid 2D semantic scene generation. *arXiv preprint arXiv:2304.01519*, 2023. 3, 7

[107] Jiayu Yang, Enze Xie, Miaomiao Liu, and Jose Alvarez. Parametric depth based feature representation learning for object detection and segmentation in bird's-eye view. In *ICCV*, 2023. 8

[108] Jingru Yi, Pengxiang Wu, Bo Liu, Qiaoying Huang, Hui Qu, and Dimitris Metaxas. Oriented object detection in aerial images with box boundary-aware vectors. In *WACV*, 2021. 6

[109] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D object detection and tracking. In *CVPR*, 2021. 3

[110] Xiang Yu, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *RSS*, 2018. 1

[111] Syed Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for fast image restoration and enhancement. *TPAMI*, 2022. 19, 20

[112] Hao Zhang, Hongyang Li, Xingyu Liao, Feng Li, Shilong Liu, Lionel Ni, and Lei Zhang. DA-BEV: Depth aware BEV transformer for 3D object detection. *arXiv preprint arXiv:2302.13002*, 2023. 3

[113] Jinqing Zhang, Yanan Zhang, Qingjie Liu, and Yunhong Wang. SA-BEV: Generating semantic-aware bird's-eye-view feature for multi-view 3D object detection. In *ICCV*, 2023. 8

[114] Renrui Zhang, Han Qiu, Tai Wang, Xuanzhuo Xu, Ziyu Guo, Yu Qiao, Peng Gao, and Hongsheng Li. MonoDETR: Depth-guided transformer for monocular 3D object detection. In *ICCV*, 2023. 5, 6, 7, 19, 22

[115] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3D object detection. In *CVPR*, 2021. 3

[116] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. BEVerse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022. 1, 3, 5, 6, 7, 8, 17, 18, 21

[117] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, 2022. 3

[118] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang. MonoEF: Extrinsic parameter free monocular 3D object detection. *TPAMI*, 2021. 3

[119] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3D object detection. In *CVPR Workshop*, 2019. 1, 8

[120] Zijian Zhu, Yichi Zhang, Hai Chen, Yinpeng Dong, Shu Zhao, Wenbo Ding, Jiachen Zhong, and Shibao Zheng. Understanding the robustness of 3D object detection with bird's-eye-view representations in autonomous driving. In *CVPR*, 2023. 1

[121] Zhuofan Zong, Dongzhi Jiang, Guanglu Song, Zeyue Xue, Jingyong Su, Hongsheng Li, and Yu Liu. Temporal enhanced training of multi-view 3D object detector via historical object prediction. In *ICCV*, 2023. 1, 3, 6, 7, 8, 17, 18, 21