LeAD-M3D: Leveraging Asymmetric Distillation for Real-Time Monocular 3D Detection

Johannes Meier [1,2,3,4,†,*], Jonathan Michel [3,4,†], Oussema Dhaouadi [1,2,3,4,5], Yung-Hsu Yang [2], Christoph Reich [3,4,6,7], Zuria Bauer [2], Stefan Roth [6,7,8], Marc Pollefeys [2,9], Jacques Kaiser [1], and Daniel Cremers [3,4,7]

[1] DeepScenario [2] ETH Zurich [3] TU Munich [4] MCML
[5] University of Cambridge [6] TU Darmstadt [7] ELIZA [8] hessian.AI [9] Microsoft
https://deepscenario.github.io/LeAD-M3D/

Abstract. Real-time monocular 3D object detection remains challenging due to severe depth ambiguity, viewpoint shifts, and the high computational cost of 3D reasoning. Existing approaches either rely on LiDAR or geometric priors to compensate for missing depth or sacrifice efficiency to achieve competitive accuracy. We introduce LeAD-M3D, a monocular 3D detector that achieves state-of-the-art accuracy and real-time inference without extra modalities. Our method is enabled by three key components. Asymmetric Augmentation Denoising Distillation (A2D2) transfers geometric knowledge from a clean-image teacher to a MixUp-noised student via a quality- and importance-weighted depth-feature loss, enabling stronger depth reasoning without LiDAR. 3D-aware Consistent Matching (CM_{3D}) improves prediction-to-ground truth assignment by integrating 3D MGIoU into the matching score, yielding stable and precise supervision. Finally, Confidence-Gated 3D Inference (CGI_{3D}) accelerates inference by restricting expensive 3D regression to confident regions. Together, these contributions set a new Pareto frontier for monocular 3D detection: LeAD-M3D achieves state-of-the-art accuracy on KITTI and Waymo, and the best reported car AP on Rope3D, while running up to 3.6× faster than prior high-accuracy models (*e.g.*, MonoDiff). LeAD-M3D demonstrates that high fidelity and real-time monocular 3D detection is simultaneously attainable, without LiDAR, stereo, or strong geometric assumptions.

Keywords: 3D Detection · Monocular · Knowledge Distillation

1 Introduction

Monocular 3d object detection (M3D) aims to predict the 3D position, orientation, and size of a scene's objects from a *single* RGB image. This setup is broadly

* j.meier@tum.de † Equal contribution

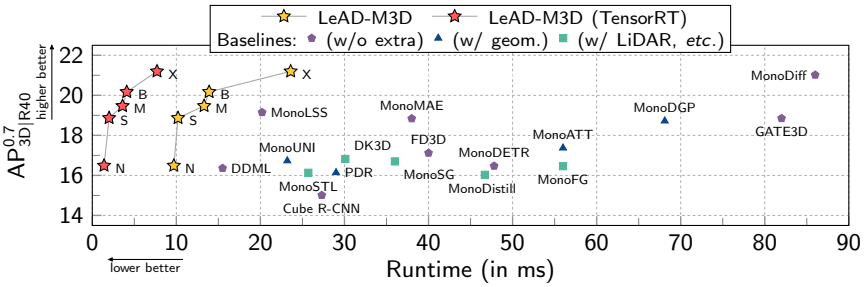

Fig. 1: Runtime *vs.* Accuracy on KITTI *test*, using $AP^{0.7}_{3D|R40}$ Mod. (in %, ↑) and runtime (in ms, ↓). We provide a model family (sizes N to X) to balance runtime and 3D detection accuracy. LeAD-M3D offers a Pareto frontier over existing approaches. Using TensorRT further improves the runtime, enabling >60 FPS real-time inference of even our largest model size (X). Runtime is reported on the same hardware (NVIDIA RTX 8000) wherever possible (*i.e.*, code is publicly available).

available and M3D finds widespread application in autonomous driving [7, 12], robotics [39], medicine [1], and city infrastructure [67]. However, inferring depth information from a single 2D image introduces strong ambiguities, and is the dominant source of error in M3D [56, 82]. This problem intensifies for non-zero roll and pitch viewpoints (*e.g.*, roadside camera views) [59, 88]. Moreover, recent methods emphasize accuracy over runtime efficiency [48, 64], hindering deployment in real-world scenarios. A model that learns exclusively from 3D box supervision, providing accurate detections and efficient real-time inference, remains currently lacking.

To obtain both accurate and efficient models, knowledge distillation (KD) has been shown to be an effective paradigm [23, 57]. In KD, a large and computationally expensive teacher model transfers its knowledge to a smaller, more efficient student model [23, 57]. This process yields a student model that retains high accuracy while significantly reducing computational cost. To perform effective KD for M3D, existing approaches create a teacher-student asymmetry by giving the teacher access to additional information, typically in the form of LiDAR [14, 27, 48, 83]. However, LiDAR is not always available in real-world applications, and the use of multiple modalities (*e.g.*, imagery and LiDAR) leads to complex pipelines. Here, we propose Asymmetric Augmentation Denoising Distillation (A2D2), an image-only KD pipeline that leverages an asymmetric denoising distillation [26] strategy for efficient and accurate M3D. Unlike prior M3D distillation pipelines, A2D2 is conceptually simpler as we do not rely on additional modalities (*e.g.*, LiDAR).

Using our A2D2, we build an M3D model family composed of five models (*cf*. Fig. 1) with different model sizes, *i.e.*, N / S / M / B / X, following the YOLO family [75]. In particular, we train our largest variant (X) as a teacher model to provide expressive target representations. We freeze the teacher and distill different student models using A2D2 by denoising a MixUp-based [90]

objective. Denoising is performed on object-specific depth features. By using a dynamic feature loss that weights features according to the teacher's prediction quality and importance, we allow for effective feature distillation. This enables the student to focus learning on reliable and informative cues, improving the student's depth reasoning and detection accuracy.

To complement our A2D2, we introduce two additional components. *First*, we introduce 3D-aware Consistent Matching (CM_{3D}), which improves the prediction-to-ground truth assignment by incorporating a 3D Intersection over Union (IoU) term into the matching criterion. This joint 2D-3D alignment allows the model to learn from supervision that better reflects the actual 3D localization quality. *Second*, we propose Confidence-Gated 3D Inference (CGI_{3D}), a lightweight inference strategy that restricts the costly 3D regression head to high-confidence regions identified by an efficient 2D classifier. This reduces redundant computation and accelerates inference with virtually no loss in accuracy.

Together, we present **LeAD-M3D**, **Le**veraging **A**symmetric **D**istillation for Real-time **M**onocular **3**D **D**etection. Our LeAD-M3D model family yields a new accuracy-efficiency Pareto frontier over existing M3D models (*cf*. Fig. 1) *without* relying on LiDAR, stereo, or ground-plane inputs. For example, on KITTI [22] we achieve the highest AP_{3D} across all difficulty levels among monocular methods and even surpass LiDAR- and geometry-assisted approaches, while running 3.6× faster than the next-best high-accuracy model (MonoDiff [64]). LeAD-M3D also achieves strong cross-view and domain generalization results, all while running in real-time on a single GPU.

Our main contributions can be summarized as follows: *(i)* We propose A2D2, a novel, LiDAR-free knowledge-distillation scheme for M3D. A2D2 transfers knowledge using MixUp-based information asymmetry and a quality- and importance-weighted feature loss, strengthening the student's depth reasoning without requiring privileged depth information. *(ii)* We introduce CM_{3D}, a prediction-to-ground truth assignment strategy that improves supervision by employing 3D cues. *(iii)* We propose CGI_{3D}, a lightweight inference strategy that restricts costly 3D regression to high-confidence regions, effectively reducing inference efficiency without a degradation in detection accuracy. *(iv)* We demonstrate a new accuracy-efficiency Pareto frontier (*cf*. Fig. 1) of LeAD-M3D across a wide range of datasets, without relying on LiDAR, stereo, or geometric priors (*e.g.*, inverse height-depth consistency). Additionally, LeAD-M3D leads to strong cross-view and domain generalization results.

2 Related Work

Monocular 3D Object Detection (M3D). Existing M3D methods can be broadly grouped into *three* categories. The *first* uses extra data, such as LiDAR [27, 61, 65, 80, 81, 87], object shape cues [40, 51], temporal consistency [11], ground planes [9, 69], or stereo images [19], during training to compensate for sparse 3D box supervision. The dependency on additional data limits the applicability to real-world scenarios where these modalities are unavailable, *e.g.*, LiDAR

is unavailable in certain traffic [73, 97] or drone [16, 59] scenarios. While LiDAR is often used to aid the annotation process, recent work removed LiDAR entirely from the M3D annotation pipeline [16]. The *second* category of M3D models injects geometric constraints to regularize depth [31, 46, 53, 63, 82, 93, 96]. A common assumption is that the apparent 2D height of an object is inversely proportional to its depth, which yields a scale prior by relating the estimated 3D height to the 2D height [53, 93]. Current approaches [31, 63, 82, 96] encode variants of perspective geometric priors. These assumptions largely hold under car-view settings with near-zero roll and pitch but break for different viewpoints, restricting deployment to forward-facing automotive cameras [59, 88, 97]. The *third* category relies only on 3D bounding boxes for supervision without auxiliary modalities or hard-coded priors [20, 28, 29, 32, 43]. We focus on this category of models, maximizing applicability across camera viewpoints and datasets. Our approach fits this family, as it allows full flexibility with respect to object orientation, explicitly supporting arbitrary SO(3) rotations rather than being constrained by viewpoint-specific priors like the inverse height-depth consistency.

Knowledge Distillation (KD) [6, 23, 57, 68] is a general technique for transferring knowledge from a large, high-capacity, and accurate teacher model to a smaller, more efficient student model. Introducing asymmetry into the distillation process has been shown to improve the student's accuracy [6, 23, 57, 68]. Previous M3D methods achieve this teacher-student asymmetry by giving the teacher an easier problem and/or more informative data, *e.g.*, supplying LiDAR, allowing the monocular student to learn high-quality 3D geometric cues [14, 27, 48, 83]. On the other hand, ADD [80] and FD3D [78] remove the LiDAR dependency by providing the teacher with ground-truth object positions or object-wise depth maps. Our approach takes a different angle on this paradigm. Instead of giving the teacher more information, we give the student strongly augmented MixUp images [90], while the teacher receives both clean images. Compared to existing M3D distillation approaches, our approach is conceptually simpler, as we do not use multiple modalities. This also allows both student and teacher to share the same meta-architecture, reducing overall architectural complexity.

Real-Time Object Detection. Significant progress has been made in 2D real-time detection [8, 35, 38, 54, 62, 74, 77, 94] on the accuracy-efficiency frontier. Among real-time 2D detection approaches, the YOLO series [35, 38, 74] has continuously improved the accuracy-efficiency frontier. We base our real-time M3D model on YOLOv10 [75], which enables accurate, efficient, and non-maximum suppression-free inference. Existing M3D methods require long [61, 65, 81] or moderate inference times [43, 49], and most provide only a single model size [43, 53, 82]. To address this, we offer a model family with five model sizes where accuracy is improved through A2D2 and task-aligned learning without increasing inference cost, while CGI_{3D} optimizes runtime. This allows our second-largest model (B) to outperform the previous fastest baseline in terms of speed.

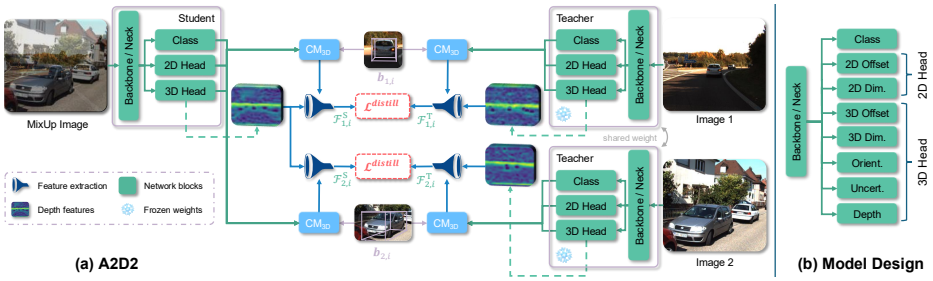

Fig. 2: Overview of LeAD-M3D. *(a)* We distill high-dimensional instance-depth features from a large teacher to a compact student. To create an information gap, the teacher sees clean images. The student receives a MixUp image and must reproduce the teacher's intermediate features (Sec. 3.1). This frames distillation as a denoising task, which removes MixUp-induced artifacts. CM_{3D} uses ground truth to pair corresponding teacher and student predictions (*cf*. Sec. 3.2). *(b)* Detailed model architecture. *Dim.* stands for dimension head, *Orient.* denotes for orientation head, and *Uncert.* stands the depth uncertainty head.

3 Method: Accurate & Real-Time M3D

Problem Statement. The goal of M3D is to predict 3D bounding boxes and object categories from a single RGB image. Formally, let $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ be an RGB image and let $\hat{B}(\mathbf{I}) = \{\hat{\mathbf{b}}_1^{\text{3D}}, \ldots, \hat{\mathbf{b}}_{M'}^{\text{3D}}\}$ denote the set of estimated 3D bounding boxes for objects in $\mathbf{I}$. Each box $\hat{\mathbf{b}}_i^{\text{3D}}$ is defined by its 3D center location $(x_i, y_i, z_i) \in \mathbb{R}^3$, 3D size $(w_i, h_i, l_i) \in \mathbb{R}^3$, orientation represented by a rotation matrix $R_i \in \text{SO}(3)$, and category $c_i \in \mathbb{C}$ (*e.g.*, "Vehicle" or "Pedestrian").

Overview. Prior M3D methods largely optimize accuracy, while runtime efficiency is often secondary. We address this by introducing LeAD-M3D, **Le**veraging **A**symmetric **D**istillation for Real-Time **M**onocular **3**D **D**etection, a novel M3D method built on YOLOv10 [75]. As shown in Fig. 2b, we follow [43] and extend YOLOv10 with standard 3D detection heads (*i.e.*, 3D offset, 3D dimension, orientation, depth, and uncertainty heads), a postprocessing pipeline, and 3D losses. This baseline establishes an efficient architectural foundation for M3D, which we denote as YOLOv10-M3D. We show more details in the supplementary material. Our overall pipeline (*cf*. Fig. 2a) further extends the baseline M3D model with three proposed components. A2D2 enhances standard KD techniques by integrating a novel augmentation-based denoising task. CM_{3D} lifts the standard 2D prediction-to-ground truth assignment to 3D space. Finally, CGI_{3D} reduces head FLOPs during inference without impacting the predictions.

3.1 Asymmetric Augmentation Denoising Distillation (A2D2)

KD is an effective approach to balance accuracy with real-time requirements. KD methods for M3D often create an information asymmetry, easing the teacher's

task and/or complicating the student's task. As depth estimation is the core bottleneck in M3D, these methods [14, 27, 83] simplify the teacher's task enormously by adding LiDAR [14,27,78,80,83]. This further strengthens the transfer for M3D yet introduces a modality dependency. Moreover, standard KD losses transfer features uniformly [14, 27, 78, 80, 83], ignoring variation in the teacher's quality and the varying influence of feature channels on the final depth. To address these issues, we propose Asymmetric Augmentation Denoising Distillation (A2D2), a LiDAR-free KD scheme that couples an asymmetric MixUp-based denoising task with quality- and importance-weighted feature alignment.

MixUp-Based Information Asymmetry. Unlike previous methods, we use MixUp [43, 59] to augment the student input, enabling the student to learn to match the teacher's instance-depth features despite augmentation, creating a desired asymmetry without requiring extra modalities. In particular, we blend two images at the pixel level and require the student to detect all objects for both images. Unlike most other augmentation strategies, MixUp preserves the full spatial extent of all objects in both images while still creating a denoising task, so no GT annotations are lost. Crucially, MixUp preserves object geometry in image coordinates, *i.e.*, projected centers, depths, dimensions, and orientation remain consistent. We exploit this invariance for distillation as a denoising task in feature space.

As shown in Fig. 2a, first, we feed two clean images to the teacher and the corresponding MixUp image to the student. Second, we use 3D-aware Consistent Matching ($\mathrm{CM}_{\mathrm{3D}}$) to assign each ground-truth object to its best-matching prediction from the teacher and separately to its best match from the student (Sec. 3.2). It is worth noting that we distill the depth features from the depth head rather than generic backbone activations, where the scalar depth is obtained by regression using the teachers weight matrix $W^{\mathrm{T}} \in \mathbb{R}^{Q}$ and the teachers depth features $\mathcal{F}_i^{\mathrm{T}} \in \mathbb{R}^{Q}$. Where i denotes a specific object in the image. This establishes explicit student-teacher pairs for the same underlying object, enabling feature-level distillation between corresponding predictions. We found that depth features yield more effective transfer for the 3D task than generic backbone features.

Quality- and Importance-Weighted Distillation Loss. The standard KD losses ignore variation in teacher quality and the unequal importance of feature channels. To address this, we propose reweighting the feature loss based on the teacher's prediction quality and the importance of each feature. For the teacher's quality, we weight the distillation loss by a relative depth-error-based score:

$$\eta_i = \frac{z_i}{\max\big(|z_i - \hat{z}_i^{\mathrm{T}}|,\ \epsilon\big)}, \tag{1}$$

where z_i and $\hat{z}_i^{\mathrm{T}}$ denote ground truth and teacher-predicted depth, respectively, for object i. We set $\epsilon = 0.1$ for numerical stability. Using a relative error mitigates

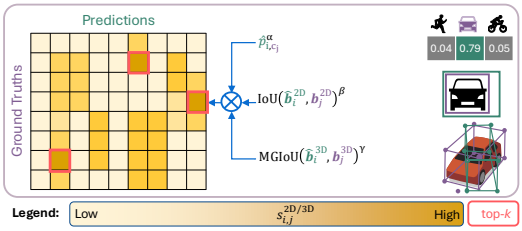

Fig. 3: 3D-aware Consistent Matching ($\mathbf{CM_{3D}}$) utilizes both 2D and 3D overlaps alongside classification scores to disambiguate prediction-to-ground truth assignments, enabling stable and precise supervision during training and distillation.

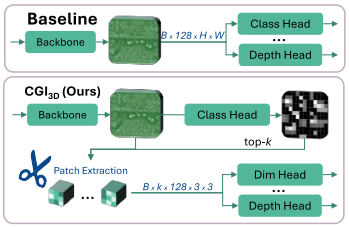

Fig. 4: Confidence-Gated 3D Inference ($\mathbf{CGI_{3D}}$) speeds up inference by restricting 2D/3D regression to confident locations (*bottom*), while the baseline processes every location (*top*).

a higher weighting of nearby over distant objects, as nearby objects naturally exhibit smaller absolute errors. For feature importance, channels with larger absolute weights contribute more to the final prediction. Therefore, we define the normalized importance per channel $q \in \{1, \dots, Q\}$ as

$$\omega_q = \frac{|W_q^{\mathrm{T}}|}{\sum_{q'=1}^{Q} |W_{q'}^{\mathrm{T}}|}. \tag{2}$$

We combine these weights with an L1 alignment on instance-depth features. Let $B(\mathbf{I}) = \{\mathbf{b}_1^{\mathrm{3D}}, \dots, \mathbf{b}_M^{\mathrm{3D}}\}$ be the ground-truth bounding box set of $\mathbf{I}$ matched to the predictions by $\mathrm{CM_{3D}}$. We define our quality- and importance-weighted feature-loss between the teacher $\mathcal{F}_i^{\mathrm{T}}$ and student features $\mathcal{F}_i^{\mathrm{S}} \in \mathbb{R}^Q$ as:

$$\mathcal{L}^{\mathrm{distill}} = \frac{1}{|B(\mathbf{I})|} \sum_{i=1}^{|B(\mathbf{I})|} \sum_{q=1}^{Q} \omega_q \, \eta_i \left| \mathcal{F}_{i,q}^{\mathrm{T}} - \mathcal{F}_{i,q}^{\mathrm{S}} \right|. \tag{3}$$

Training. We adopt offline KD [26] with a frozen teacher. Regardless of the student model size, the teacher is always uses our largest size (*i.e.*, LeAD-M3D X trained *w/o* A2D2). We first train the teacher using standard supervision [75], *i.e.*, classification $\mathcal{L}^{\mathrm{cls}}$, 2D bounding-box $\mathcal{L}^{\mathrm{2D}}$, and 3D bounding-box $\mathcal{L}^{\mathrm{3D}}$ loss, then freeze for distillation. During distillation, we train a student model, which can be any model size from our family (N, S, M, B, or X), using the total loss

$$\mathcal{L} = \mathcal{L}^{\mathrm{cls}} + \mathcal{L}^{\mathrm{2D}} + \mathcal{L}^{\mathrm{3D}} + \mathcal{L}^{\mathrm{distill}}, \tag{4}$$

composed of our distillation loss $\mathcal{L}^{\mathrm{distill}}$ and the standard supervised losses. We provide more details on the training and loss functions in the supplement.

3.2 3D-aware Consistent Matching ($\mathbf{CM_{3D}}$)

Both supervised training and our A2D2 method require reliable assignments between model predictions and ground-truth objects. Without robust matching,

noisy or misaligned pairs can degrade both training stability and final accuracy, especially in challenging scenarios like MixUp, where multiple objects with similar 2D projections coexist. To address this, we propose 3D-aware Consistent Matching ($\mathrm{CM_{3D}}$), which ranks prediction-ground truth pairs using class confidence and both 2D as well as 3D box agreement (*cf*. Fig. 3).

2D matches in our baseline (YOLOv10-M3D [75]) are obtained by ranking all prediction-ground truth pairs using the score $s_{i,j}^{\mathrm{2D}}$, and top-k selection. $s_{i,j}^{\mathrm{2D}}$ is computed by:

$$s_{i,j}^{\mathrm{2D}} = \hat{p}_{i,c_j}^{\alpha} \ \mathrm{IoU}\big(\hat{\mathbf{b}}_i^{\mathrm{2D}}, \mathbf{b}_j^{\mathrm{2D}}\big)^{\beta}, \tag{5}$$

where i indexes predictions, j indexes ground-truth bounding boxes, and $\hat{p}_{i,c_j}$ is the predicted probability for class c_j. $\hat{\mathbf{b}}_i^{\mathrm{2D}}$ and $\mathbf{b}_j^{\mathrm{2D}}$ are the predicted and ground-truth 2D bounding boxes. $\alpha \in \mathbb{R}_+$ and $\beta \in \mathbb{R}_+$ are weighting bounding box overlap and classification. We extend this to 3D by incorporating the Marginalized Generalized IoU (MGIoU) [37] between predicted and ground-truth 3D bounding boxes (*cf*. Fig. 3):

$$s_{i,j}^{\mathrm{2D/3D}} = s_{i,j}^{\mathrm{2D}} \ \mathrm{MGIoU}\big(\hat{\mathbf{b}}_i^{\mathrm{3D}}, \mathbf{b}_j^{\mathrm{3D}}\big)^{\gamma}, \tag{6}$$

with the predicted $\hat{\mathbf{b}}_i^{\mathrm{3D}}$ and the ground-truth $\mathbf{b}_j^{\mathrm{3D}}$ 3D bounding boxes. $\gamma \in \mathbb{R}_+$ is used for weighting. MGIoU offers a 3D-overlap surrogate that remains informative even without 3D box intersection, which is common early in training or for small objects. It is computed by projecting the 3D shapes onto a set of unique directional normals and marginalizing the one-dimensional generalized IoU values across these projections. MGIoU aggregates location, size, and orientation, and is invariant to object dimensions, unlike corner-based losses [46]. Note, using MGIoU leads to a negligible increase in training time by less than 2 %.

Compared to static anchor-based assignments used in many monocular 3D object detection (M3D) methods [49, 59], our dynamic 2D/3D scoring better disambiguates crowded scenes and mixed content (*e.g.*, under MixUp). The 2D term stabilizes learning when 3D estimates are crude, while the 3D term helps to separate overlapping predictions as accuracy improves.

3.3 Confidence-Gated 3D Inference ($\mathbf{CGI_{3D}}$)

M3D detectors typically run their regression heads densely over full feature maps. As most locations correspond to background, a substantial portion of the computation is wasted. Therefore, we propose Confidence-Gated 3D Inference ($\mathrm{CGI_{3D}}$) to restrict expensive regression to likely object locations as shown in Fig. 4. Specifically, *(1)* after obtaining the neck features, we run the classification head densely across the entire feature map to *(2)* select the top-k locations based on class confidence. We then *(3)* extract 3×3 local patches centered at these locations and *(4)* apply the 2D and 3D regression heads only to these sparse patches rather than the full map.

During training, we keep the dense head computation for simplicity. At inference, we move the baseline top-k selection earlier and run the 2D/3D heads

only on 3×3 patches around top-k locations, yielding outputs identical to dense evaluation, because the heads' effective receptive field is exactly 3×3 (one 3×3 convolution followed by two 1×1 convolutions). This is simpler than Region of Interest (RoI)-Align (typically 7×7 grids [24, 66]) and avoids bilinear interpolation, substantially reducing head-level FLOPs with no accuracy loss.

4 Experiments

We first compare our approach against lightweight and state-of-the-art (SOTA) M3D methods, followed by domain generalization evaluations. Finally, we analyze individual model components and provide qualitative results.

Datasets and Metrics. We perform experiments on the primary M3D benchmarks, KITTI [22] and Waymo [71]. To evaluate diverse camera perspectives, we conduct experiments on Rope3D [88]. Rope3D includes different cameras and viewpoints, enabling cross-view validation. To showcase domain generalization, we use nuScenes (Front) [5]. We strictly follow standard protocols and metrics across all datasets. In particular, for KITTI [10], we report $AP_{3D|R40}$ and $AP_{BEV|R40}$ at IoU thresholds of 0.7 ("Cars") and£ 0.5 ("Pedestrians"/"Cyclists"). Both metrics are computed for easy, moderate (Mod.), and hard objects. Note that while our main experiments are performed on the KITTI *test* set (requiring a submission to the official test server), analysis and ablations are performed on the *validation* set. Waymo evaluation follows the DEVIANT [36] protocol for front-camera images (AP_{3D}/APH_{3D} at 0.5/0.7 IoU). For Rope3D, we use the heterologous split and report $AP_{3D|R40}$ and the Rope score [88] for "Car" and "Big Vehicle" classes. To measure inference speed and efficiency, we report model parameters, runtime, and floating point operations (FLOPs). Both runtime and FLOPs are reported on the same hardware (NVIDIA RTX 8000), if code is available, and for a single image forward pass.

Implementation Details. We train our student and teacher models using the Adam optimizer [34] with an initial learning rate of 0.001, a weight decay of 0.0005, and a 3-epoch warmup cosine learning rate schedule [52]. CM_{3D} hyperparameters are set to $\alpha = 0.5$, $\beta = 1.0$, and $\gamma = 1.0$. All experiments are conducted on a single NVIDIA RTX 8000. The teacher training (LeAD-M3D w/o A2D2 requires 34 hours; subsequent student training scales with model size, peaking at 60 h for the largest model (size X) on KITTI. Despite the focus on optimized inference speed for real-world applications, our training remains resource-efficient, requiring only a *single* GPU. As our direct baseline, we use YOLOv10 by adding 3D detection heads (YOLOv10-M3D). For all implementation details, please refer to the supplement.

4.1 Comparison with Lightweight M3D Models

In Tab. 1, we compare our model with existing lightweight M3D models (<30 M parameters) in terms of accuracy, FLOPs, model size, and runtime. Mono-

Table 1: Comparison with lightweight M3D methods (< 30 M parameters) on the KITTI [22] *test* set for the category "Car" using $AP^{0.7}_{3D|R40}$ and $AP^{0.7}_{BEV|R40}$ (both in %, ↑). *Extra* indicates the use of auxiliary training data. *Params* reports no. of model parameters in millions. *GFLOPs* measured for single image inference. *Time* is reported in ms for single image inference without TensorRT on an NVIDIA RTX 8000 GPU. Best and second-best results are highlighted in blue and light blue, respectively.

Method	Extra	Params ↓	GFLOPs ↓	Time ↓	$AP^{0.7}_{3D\|R40}$ ↑			$AP^{0.7}_{BEV\|R40}$ ↑		
					Easy	Mod.	Hard	Easy	Mod.	Hard
MonoNERD [81]	LiDAR	6.6	4220	1380.3	22.75	17.13	15.63	31.13	23.46	20.97
MonoSGC [30]	LiDAR	23.1	173	35.0	27.01	16.77	14.61	35.78	23.27	19.92
OccupancyM3D [61]	LiDAR	28.3	389	213.9	25.55	17.02	14.79	35.38	24.18	21.37
DPL_{FLEX} [91]	Unlabeled	21.5	152	28.9	24.19	16.67	13.83	33.16	22.12	18.74
MonoUNI [31]	Geometry	22.7	122	23.2	24.75	16.73	13.49	33.28	23.05	16.39
MonoCD [82]	Geometry	21.8	171	28.1	25.53	16.59	14.53	33.41	22.81	19.57
MonoCon [49]	—	19.6	115	15.5	22.50	16.46	13.95	31.12	22.10	19.00
DDML [13]	—	19.6	115	15.5	23.31	16.36	13.73	—	—	—
MonoLSS [43]	—	21.5	127	20.2	26.11	19.15	16.94	34.89	25.95	22.59
LeAD-M3D N (Ours)	—	3.8	14	9.7	24.31	16.49	14.14	32.22	21.72	19.41
LeAD-M3D S (Ours)	—	10.1	38	10.2	27.28	18.87	16.37	34.86	24.17	21.32
LeAD-M3D M (Ours)	—	19.7	88	13.3	28.08	19.47	17.66	36.21	25.46	22.89
LeAD-M3D B (Ours)	—	24.9	133	13.9	29.10	20.17	18.34	37.65	26.63	23.75

NeRD [81] uses fewer parameters than any prior method (6.6 M). Despite this, our second-smallest model (S) surpasses MonoNeRD in accuracy while being trained LiDAR-free. Notably, LeAD-M3D S has an over 100× faster inference than MonoNeRD. We attribute MonoNeRD's low speed to its compute-heavy 3D volume processing, which dominates its inference cost. Among models trained without extra data, MonoLSS is the most accurate existing approach. Still, LeAD-M3D B outperforms MonoLSS on almost all accuracy metrics, while being 34 % faster in runtime. In terms of FLOPs, our third-largest model (M) requires 25 % fewer FLOPs than the fastest alternatives, DDML [13] and MonoCon [49], yet outperforms all existing models on 5 out of 6 accuracy metrics. Our second-largest model (B) maintains a faster runtime than the fastest existing lightweight baseline while surpassing all prior methods in accuracy.

4.2 Comparison with SOTA M3D Models

KITTI [22] Test Set. In Tab. 2, we compare our largest model LeAD-M3D X with SOTA models on the KITTI test set (requiring submission to the test server). LeAD-M3D outperforms all existing methods in $AP_{3D|R40}$, including those that leverage perspective geometric priors (*e.g.*, inverse height-depth consistency) or extra training data. Furthermore, our model runs 3.6 × faster than the previous SOTA MonoDiff [64] (*cf.* supplement).

Rope3D [88] Validation Set. In Tab. 3, we provide results on Rope3D [88] (traffic view), evaluating accuracy for views different from car mounded cameras. LeAD-M3D X scores the best result for the "Car" category. For the rare category

Table 2: KITTI [22] test results. Comparison with state-of-the-art M3D methods on the KITTI *test* set for the category "Car" using $AP^{0.7}_{3D|R40}$ (in %). *Extra* indicates the use of auxiliary train. data.

Method	Extra	$AP^{0.7}_{3D\|R40}$ ↑		
		Easy	Mod.	Hard
MonoDistill [14]	LiDAR	24.31	18.47	15.76
ADD [80]	LiDAR	25.61	16.81	13.79
HSRDN [83]	LiDAR	22.01	13.61	13.10
MonoFG [21]	LiDAR	24.35	16.46	13.84
MonoSTL [17]	LiDAR	25.33	16.13	13.35
MonoSG [19]	Stereo	25.77	16.70	14.22
DK3D [79]	LiDAR	25.63	16.82	13.81
MonoTAKD [48]	LiDAR	27.91	19.43	16.51
MoGDE [96]	Geometry	27.25	17.93	15.80
MonoDGP [63]	Geometry	26.35	18.72	15.97
Cube R-CNN [2]	—	23.59	15.01	12.56
MonoDETR [92]	—	25.00	16.47	16.38
MonoPSTR [85]	—	26.15	17.01	13.70
FD3D [78]	—	25.38	17.12	14.50
MonoDiff [64]	—	30.18	21.02	18.16
MonoMAE [32]	—	25.60	18.84	16.78
GATE3D [29]	—	26.07	18.85	16.76
MonoA^2 [18]	—	23.24	17.55	15.26
LeAD-M3D X (Ours)	—	30.76	21.20	18.76

Table 3: Rope3D [88] results. Comparison with with state-of-the-art M3D methods on the Rope3D heterlog. benchmark. GPF indicates ground-plane-free models. We report $AP^{0.7}_{3D|R40}$ and Rope score in %. BV denotes "Big Vehicle" class.

Method	GPF	Car ↑		BV ↑	
		AP	Rope	AP	Rope
M3D-RPN [3]	✗	11.09	28.17	3.39	21.01
MonoDLE [56]	✗	12.16	28.39	3.02	19.96
MonoFlex [93]	✗	11.24	27.79	13.10	28.22
BEVHeight [86]	✗	5.41	23.09	1.16	18.53
CoBEV [69]	✗	6.59	24.01	2.26	19.71
M3D-RPN [3]	✓	6.05	23.84	2.78	20.82
Kinematic3D [4]	✓	5.82	23.06	1.27	18.92
MonoDLE [56]	✓	3.77	21.42	2.31	19.55
MonoFlex [93]	✓	10.86	27.39	0.97	18.18
BEVFormer [45]	✓	3.87	21.84	0.84	18.42
BEVDepth [41]	✓	0.85	19.38	0.30	17.84
MonoCon [49]	✓	10.71	27.55	1.61	19.25
GroundMix [59]	✓	12.86	29.37	3.90	21.06
LeAD-M3D N (Ours)	✓	9.67	25.46	1.80	19.17
LeAD-M3D S (Ours)	✓	13.33	28.59	4.16	21.25
LeAD-M3D M (Ours)	✓	14.31	29.62	4.95	22.14
LeAD-M3D B (Ours)	✓	15.05	30.13	5.40	22.41
LeAD-M3D X (Ours)	✓	16.45	31.34	8.71	25.15

"Big Vehicle", it performs second-best to MonoFlex [93], which requires ground-plane inputs, while LeAD-M3D does not. In a fair comparison (*i.e.*, MonoFlex w/o ground-plane input), LeAD-M3D significantly outperforms MonoFlex.

Waymo [71] Validation Set. Similarly, our largest model (X) achieves the best results on the Waymo validation set [71] (*cf.* Tab. 4) without perspective geometric priors or extra training data. In $AP^{0.5}_{3D}$ Level 1, we surpass the previous best model [43] by 2.97 AP. Even our second-largest model (B) attains a higher $AP^{0.5}_{3D}$ than all existing methods.

Domain Generalization. In Tab. 5, we evaluate the robustness of LeAD-M3D B to domain shifts. In particular, we analyze domain generalization by training on nuScenes [5] and evaluating on the KITTI [22] validation set. LeAD-M3D B outperforms all domain generalization (DG) approaches, even MonoGDG [84], a specialized approach for DG, requiring alignments for focal length, distortion, and camera orientation. In contrast, LeAD-M3D B generalizes using only virtual depth [2] without specific components designed for DG. Notably, LeAD-M3D remains competitive with unsupervised domain adaptation (UDA) methods that leverage target data for adaptation. This demonstrates that LeAD-M3D learns highly generalizable representations effective in diverse sensor setups.

Table 4: Waymo [71] results. Comparison with state-of-the-art M3D methods on the Waymo *validation* set. We report $AP^{0.5}_{3D}$ & $AP^{0.7}_{3D}$ (both in %, ↑) for the "Vehicle" category. We compare with methods following the DEVIANT [36] protocol.

Method	$AP^{0.5}_{3D}$ ↑		$AP^{0.7}_{3D}$ ↑	
	L1	L2	L1	L2
PatchNet [55]	2.92	2.42	0.39	0.38
PCT [76]	4.20	4.03	0.89	0.66
GUPNet [53]	10.02	9.39	2.28	2.14
DEVIANT [36]	10.98	10.29	2.69	2.52
MonoJSG [47]	5.65	5.34	0.97	0.91
MonoCon [49]	10.07	9.44	2.30	2.16
Stereoscopic [33]	7.18	7.17	1.72	1.61
MonoRCNN++ [70]	11.37	10.79	4.28	4.05
MonoUNI [31]	10.98	10.38	3.20	3.04
MonoXiver-GUPNet [50]	11.47	10.67	—	—
MonoXiver-DEVIANT [50]	11.88	11.06	—	—
DDML [13]	10.14	9.50	2.50	2.34
MonoAux [42]	9.82	9.25	3.92	3.57
NF-DVT [60]	11.32	11.24	2.76	2.64
MonoLSS [43]	13.49	13.12	3.71	3.27
SSD-MonoDETR [25]	11.83	11.34	4.54	4.12
GroundMix [59]	11.89	10.50	3.10	2.73
MonoCD [82]	11.62	11.14	3.85	3.50
MonoDGP [63]	12.36	11.71	4.28	4.00
LeAD-M3D N (Ours)	12.14	10.73	2.96	2.61
LeAD-M3D S (Ours)	13.24	12.08	3.53	3.11
LeAD-M3D M (Ours)	14.55	12.86	3.98	3.51
LeAD-M3D B (Ours)	15.04	13.29	4.29	3.78
LeAD-M3D X (Ours)	16.46	14.54	4.82	4.24

Table 5: Domain generalization results for nuScenes [5] to KITTI [22] *val.* using $AP^{0.5}_{3D|R40}$ (in %, ↑). We compare to domain generalization (DG) and unsupervised adaptation approaches (UDA, in gray) that use target-domain data.

Method	Type	Easy	Mod.	Hard
STMono3D [44]	UDA	29.01	19.88	17.17
MonoCT [58]	UDA	42.80	32.24	27.36
DGMono3D [95]	DG	28.77	24.82	23.67
MonoGDG [84]	DG	33.48	27.14	26.37
LeAD-M3D B (Ours)	DG	45.50	32.14	28.56

Table 6: CM_{3D} & A2D2 analysis on KITTI *val.* using $AP^{0.7}_{3D|R40}$ (in %, ↑, "Car") & median depth error (MDE, in %, ↓). Configuration 1 is YOLOv10-M3D B (our baseline); config. 5 is LeAD-M3D B.

Config.	A2D2	CM_{3D}		$AP^{0.7}_{3D\|R40}$ ↑			MDE ↓
		2D	3D	Easy	Mod.	Hard	in cm
1		✓		25.68	19.60	17.47	61
2		✓	✓	26.26	20.43	17.71	60
3	✓	✓		27.12	21.81	19.34	57
4	✓		✓	26.34	21.49	19.12	57
5 (full)	✓	✓	✓	28.44	22.65	19.87	56

4.3 Analyzing LeAD-M3D

We conduct multiple analyses and ablations to demonstrate the effectiveness of our key contributions. We report results using the KITTI [22] validation and use LeAD-M3D B if not noted differently. Further analyses are in the supplement.

Main Analyses. Table 6 analyzes both A2D2 and CM_{3D} w.r.t. the detection accuracy. Starting from our baseline (YOLOv10-M3D B), adding 3D cues in CM_{3D} (config. 2) improves the AP (Moderate) by 0.83 % and the depth error by 1 cm. Adding distillation using A2D2 (config. 5) further improves the accuracy by 2.22 % in AP (Mod.) and significantly reduces the depth error by 4 cm w.r.t. configuration 2. These results demonstrate that the A2D2, our core contribution, is the key driver to strong detection and depth accuracy of LeAD-M3D. Still, both A2D2 and CM_{3D} improve the accuracy independently of each other (*cf.* config. 2, 3 & 4 in Tab. 6).

A2D2 Ablation. In Tab. 7, we ablate individual components of our A2D2 approach in detail by removing or adding specific components from our full configuration (LeAD-M3D B). Distilling backbone features instead of depth features yields a reduced AP (Mod.) by 0.35 %, confirming that depth features are

Table 7: Detailed A2D2 ablation on KITTI [22] *validation* using car category $AP^{0.7}_{3D|R40}$ (in %) and model size B. We ablate the key components of A2D2 and report results without A2D2 for reference.

Method	Easy ↑	Mod. ↑	Hard ↑
LeAD-M3D B (Ours)	28.44	22.65	19.87
w/ backbone features	27.56	22.30	19.82
w/o quality indicator	27.79	22.28	19.63
w/o importance indicator	27.33	22.04	19.37
w/ online self-distillation	27.45	21.80	19.57
w/o clean images	27.19	21.57	19.27
Ours w/o A2D2	26.26	20.43	17.71

Table 8: A2D2 augmentation analysis. We analyze different augmentation strategies for distillation on the KITTI [22] *validation* set using our B model and report Car Moderate $AP^{0.7}_{3D|R40}$ in %. For reference, we also report results without distillation (w/o A2D2).

Augmentation strategy	Car $AP^{0.7}_{3D\|R40}$ ↑
LeAD-M3D w/o A2D2	20.43
RandAugment [15]	21.10
CutMix [89]	22.28
MixUp [90]	22.65

Table 9: Runtime analysis of CGI_{3D} on KITTI [22] *validation* using car category $AP^{0.7}_{3D|R40}$ (in %), model size N, and single image inference. Ch. denotes the number of regression head channels. Time is reported in ms.

CGI_{3D}	Ch.	$AP^{0.7}_{3D\|R40}$ ↑	GFLOPs ↓	Time ↓
	128	18.33	81.0	16.2
✓	128	18.33	19.5	10.9
	64	18.38	45.8	15.5
✓	64	18.38	14.2	8.2

Table 10: Analysis of augmentation difficulty. We measure teacher (LeAD-M3D X w/o A2D2) depth error (in cm, ↓) on augmented KITTI [22] *val.* sets. MixUp induces the highest error, providing an effective asymmetry for distillation.

Augmentation strategy	Depth Error (cm) ↓	
	Mean	Median
No aug. (KITTI val.)	118	59
RandAugment [15]	129	62
CutMix [89]	137	66
MixUp [90]	167	75

a more effective KD target than generic backbone features. Removing quality and importance indicators reduces AP (Mod.) by 0.37 % and 0.61 %, respectively. Performing online, *i.e.*, distilling during initial supervised training, instead of offline distillation, reduces AP (Mod.) by 0.85 %. This is likely because the strongest distillation targets emerge only late in training, coinciding with the learning rate decay necessary for depth convergence. Finally, feeding the teacher the same MixUp image as the student, *i.e.*, removing asymmetry from augmentations, yields a drop in AP (Mod.) of 1.08 %, demonstrating that augmentation asymmetry/denoising is beneficial for effective distillation.

Runtime Analysis. In Tab. 9, we analyze the effect of CGI_{3D} on the inference efficiency and accuracy of LeAD-M3D N. In particular, we analyze our two core modifications of the baseline (YOLOv10-M3D): First, applying CGI_{3D} during inference, and second, reducing the head channels from 128 to 64. Applying CGI_{3D} significantly reduces the runtime by two-thirds and FLOPs by about 75 %, while retaining the accuracy. Reducing head channels further improves the runtime by 0.7 ms and FLOPs by ca. 40 %.The AP remains about the same. Together, our modifications significantly improve inference efficiency without sacrificing accuracy. Please see the supplement for additional results and insights.

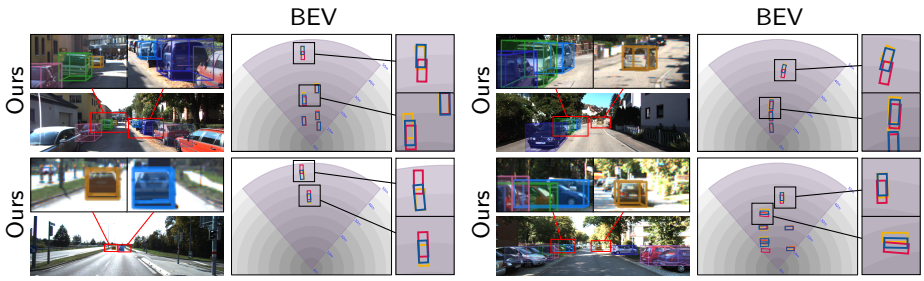

Fig. 5: Qualitative results on the KITTI [22] *val.* set. We visualize LeAD-M3D X's predictions in 2D *(left)* and compare LeAD-M3D X to our baseline (YOLOv10-M3D X) in the bird's eye view (BEV) *(right)*. LeAD-M3D X achieves more accurate depth estimates than our baseline (YOLOv10-M3D X). We highlight improved detections. Best viewed in color; zoom in for details. BEV color coding: Ground truth ■, YOLOv10-M3D X ■, LeAD-M3D X ■, and field of view ■.

Augmentation Strategies for A2D2. A2D2 enables the student to improve the accuracy by artificially complicating the student's task. A key component during this distillation is the data augmentation strategy, which introduces asymmetry and requires denoising. To analyze the effectiveness of MixUp in A2D2, we report results with different augmentation strategies in Tab. 8. In particular, we perform distillation with MixUp [90], CutMix [89], and RandAugment [15] (omitting geometric transforms). MixUp outperforms both CutMix and RandAugment. Still, all augmentation strategies improve over performing no distillation, demonstrating the general effectiveness of A2D2.

To further analyze why A2D2 is effective, we generate augmented versions of the KITTI validation set. In particular, we utilize different augmentation strategies (*i.e.*, MixUp, CutMix, and RandAugment) to generate augmented validation sets. On these augmented validation sets, we report the depth error using LeAD-M3D X *w/o A2D2* in Tab. 10. All augmentation strategies lead to an increased depth error over the error on the clean validation set (no aug.), while MixUp leads to the largest depth error. This shows that augmentations effectively introduce an information asymmetry, *i.e.*, complicating the student's objective. As demonstrated by the results in Tabs. 7 and 8, the student benefits from the large asymmetry introduced by MixUp, leading to increased detection accuracy and depth reasoning.

4.4 Qualitative Results

In Fig. 5, we show qualitative results on the KITTI [22] validation set. Compared to the baseline, LeAD-M3D's detections entail noticeably more accurate depth estimates. The improvements are particularly pronounced for moderate- and far-range objects, where estimating depth is especially challenging. For more qualitative results, please refer to the supplement.

Fig. 6: Saliency analysis. Integrated gradients [72] w.r.t. object depth (left bottom car) on KITTI *val*. Without denoising & distillation *(middle)*, saliency attends to noisy background regions. With both *(right)*, saliency concentrates on the object.

4.5 Understanding Denoising and Distillation.

Our approach combines denoising and distillation (*cf*. Fig. 2) to obtain accurate 3D detections. However, both components entail different mechanistic properties. MixUp-based denoising *increases detection difficulty* (*cf*. Tab. 10), thereby using annotations more effectively. Denoising also acts as a *regularizer*, forcing the student to rely on robust and MixUp-invariant cues, *e.g.*, object structure, rather than noisy background structure, as demonstrated qualitatively in Fig. 6. Depth feature distillation by A2D2 acts as an effective *guidance* signal for denoising (*cf*. Tab. 7). Together, denoising and distillation enable the student to improve 3D reasoning and detection accuracy.

5 Conclusion

We presented a real-time monocular 3D detection framework that employs asymmetric augmentation denoising distillation, 3D-aware matching, and confidence-gated inference. Our LeAD-M3D model family establishes a new accuracy-efficiency Pareto frontier for M3D, without using LiDAR supervision, stereo, or perspective geometric assumptions/inputs (*e.g.*, height-depth consistency assumption or ground-plane input). LeAD-M3D achieves robust accuracy across viewpoints and datasets while also demonstrating strong domain generalization. While we focus on the supervised setting, our distillation could extend to semi-supervised settings, providing an avenue for scaling M3D.

Acknowledgments. This work is a result of the joint research project STADT:up and was funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK), based on a decision of the German Bundestag. The author is solely responsible for the content of this publication. This work was also supported by the ERC Advanced Grant SIMULACRON, the Georg Nemetschek Institute project AI4TWINNING & the DFG project 4D-YouTube CR 250/26-1. Funding was also received from the Deutsche Forschungsgemeinschaft (German Research Foundation, DFG) under Germany's Excellence Strategy (EXC-3057/1 "Reasonable Artificial Intelligence", Project No. 533677015) and the LOEWE initiative (Hesse, Germany) within the emergenCITY center [LOEWE/1/12/519/03/05.001(0016)/72]. C. Reich is supported by the Konrad Zuse School of Excellence in Learning and Intelligent Systems (ELIZA) through the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education & Research. This work was also funded by the ETH Foundation Project 2025-FS-352, the SNSF Advanced Grant 216260.

References

1. Boroukhian, T., Supyen, K., Samson, J.B., Bashyal, A., Wicaksono, H.: Integrating 3D object detection with ontologies for accurate digital twin creation in manufacturing systems. Int. J. Adv. Manuf. Technol. **140**(9), 4679–4711 (2025)
2. Brazil, G., Kumar, A., Straub, J., Ravi, N., Johnson, J., Gkioxari, G.: Omni3D: A large benchmark and model for 3D object detection in the wild. In: CVPR. pp. 13154–13164 (2023)
3. Brazil, G., Liu, X.: M3D-RPN: Monocular 3D region proposal network for object detection. In: ICCV. pp. 9286–9295 (2019)
4. Brazil, G., Pons-Moll, G., Liu, X., Schiele, B.: Kinematic 3D object detection in monocular video. In: ECCV. vol. 12368, pp. 135–152 (2020)
5. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11618–11628 (2020)
6. Chen, G., Choi, W., Yu, X., Han, T.X., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: NeurIPS. vol. 30, pp. 742–751 (2017)
7. Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., Li, H.: End-to-end autonomous driving: Challenges and frontiers. IEEE Trans. Pattern Anal. Mach. Intell. **46**(12), 10164–10183 (2024)
8. Chen, Q., Su, X., Zhang, X., Wang, J., Chen, J., Shen, Y., Han, C., Chen, Z., Xu, W., Li, F., Zhang, S., Yao, K., Ding, E., Zhang, G., Wang, J.: LW-DETR: A transformer replacement to YOLO for real-time detection. arXiv:2406.03459 [cs.CV] (2024)
9. Chen, X., Chen, M., Tang, S., Niu, Y., Zhu, J.: MOSE: Boosting vision-based roadside 3D object detection with scene cues. arXiv:2404.05280 [cs.CV] (2024)
10. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R.: 3D object proposals for accurate object class detection. NIPS **28** (2015)
11. Cheng, H., Peng, L., Yang, Z., Lin, B., He, X., Wu, B.: Temporal feature fusion for 3D detection in monocular video. IEEE Trans. Image Process. **33**, 2665–2675 (2024)
12. Chib, P.S., Singh, P.: Recent advancements in end-to-end autonomous driving using deep learning: A survey. IEEE Trans. Intell. Veh. **9**(1), 103–118 (2024)
13. Choi, W., Shin, M., Im, S.: Depth-discriminative metric learning for monocular 3D object detection. In: NeurIPS. vol. 36, pp. 80165–80177 (2023)
14. Chong, Z., Ma, X., Zhang, H., Yue, Y., Li, H., Wang, Z., Ouyang, W.: MonoDistill: Learning spatial features for monocular 3D object detection. In: ICLR (2022)
15. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: RandAugment: Practical automated data augmentation with a reduced search space. In: NeurIPS. vol. 33, pp. 18613–18624 (2020)
16. Dhaouadi, O., Meier, J., Wahl, L., Kaiser, J., Scalerandi, L., Wandelburg, N., Zhou, Z., Berinpanathan, N., Banzhaf, H., Cremers, D.: Highly accurate and diverse traffic data: The deepscenario open 3D dataset. In: IV. pp. 377–384 (2025)
17. Ding, R., Yang, M., Zheng, N.: Selective transfer learning of cross-modality distillation for monocular 3D object detection. IEEE Trans. Circuits Syst. Video Technol. **34**(10), 9925–9938 (2024)
18. Dong, J., Zhou, S., Hu, Y., Huang, Y., Jiang, J., Zuo, W., Chen, S., Zheng, N.: MonoA^2: Adaptive depth with augmented head for monocular 3D object detection. Pattern Recognit. **172**, 112418 (2026)

19. Fan, Z., Xu, C., Chu, M., Huang, Y., Ma, Y., Wang, J., Xu, Y., Wu, D.: MonoSG: Monocular 3D object detection with stereo guidance. IEEE Robotics Autom. Lett. **10**(4), 3604–3611 (2025)
20. Fischer, T., Yang, Y.H., Kumar, S., Sun, M., Yu, F.: CC-3DT: Panoramic 3D object tracking via cross-camera fusion. In: CoRL (2022)
21. Gao, H., Yu, X., Xu, Y., Ran, Q., Hussain, W.: MonoFG: Monocular 3D object detection with knowledge distillation for human-centric autonomous driving systems. ACM Trans. Auton. Adapt. Syst. (2024)
22. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR. pp. 3354–3361 (2012)
23. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. Int. J. Comput. Vis. **129**(6), 1789–1819 (2021)
24. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV. pp. 2980–2988 (2017)
25. He, X., Yang, F., Yang, K., Lin, J., Fu, H., Wang, M., Yuan, J., Li, Z.: SSD-MonoDETR: Supervised scale-aware deformable transformer for monocular 3D object detection. IEEE Trans. Intell. Veh. **9**(1), 555–567 (2024)
26. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NIPS Deep Learning & Representation Learning Workshop (2014)
27. Hong, Y., Dai, H., Ding, Y.: Cross-modality knowledge distillation network for monocular 3D object detection. In: ECCV. vol. 13670, pp. 87–104 (2022)
28. Hu, H.N., Cai, Q.Z., Wang, D., Lin, J., Sun, M., Krähenbühl, P., Darrell, T., Yu, F.: Joint monocular 3D vehicle detection and tracking. In: ICCV. pp. 5390–5399 (2019)
29. Im, E., Lee, J.K., Jee, C.: GATE3D: Generalized attention-based task-synergized estimation in 3D. In: CVPRW (2025)
30. Ji, Y., Xu, J.: Depth estimation from surface-ground correspondence for monocular 3D object detection. IEEE Trans. Intell. Transp. Syst. **25**(11), 16312–16322 (2024)
31. Jia, J., Li, Z., Shi, Y.: MonoUNI: A unified vehicle and infrastructure-side monocular 3D object detection network with sufficient depth clues. In: NeurIPS. vol. 36, pp. 11703–11715 (2023)
32. Jiang, X., Jin, S., Zhang, X., Shao, L., Lu, S.: MonoMAE: Enhancing monocular 3D detection through depth-aware masked autoencoders. In: NeurIPS. vol. 37, pp. 11392–11411 (2024)
33. Kim, J.U., Kim, H., Ro, Y.M.: Stereoscopic vision recalling memory for monocular 3D object detection. IEEE Trans. Image Process. **32**, 2749–2760 (2023)
34. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
35. Kotthapalli, M., Ravipati, D., Bhatia, R.: YOLOv1 to YOLOv11: A comprehensive survey of real-time object detection innovations and challenges. arXiv:2508.02067 [cs.CV] (2025)
36. Kumar, A., Brazil, G., Corona, E., Parchami, A., Liu, X.: DEVIANT: Depth equivariant network for monocular 3D object detection. In: ECCV. vol. 13669, pp. 664–683 (2022)
37. Le, D.T., Pham, T., Cai, J., Rezatofighi, H.: Marginalized generalized IoU (MGIoU): A unified objective function for optimizing any convex parametric shapes. arXiv:2504.16443 [cs.CV] (2025)
38. Lei, M., Li, S., Wu, Y., Hu, H., Zhou, Y., Zheng, X., Ding, G., Du, S., Wu, Z., Gao, Y.: YOLOv13: Real-time object detection with hypergraph-enhanced adaptive visual perception. arXiv:2506.17733 [cs.CV] (2025)

39. Li, M., Ma, N.: Overview of 3D object detection for robot environment perception. In: ICIVIS. pp. 675–681 (2023)
40. Li, Y., Chen, Y., He, J., Zhang, Z.: Densely constrained depth estimator for monocular 3D object detection. In: ECCV. vol. 13669, pp. 718–734 (2022)
41. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: BEVDepth: Acquisition of reliable depth for multi-view 3D object detection. In: AAAI. pp. 1477–1485 (2023)
42. Li, Z., Zheng, W., Yang, L., Ma, L., Zhou, Y., Peng, Y.: MonoAux: Fully exploiting auxiliary information and uncertainty for monocular 3D object detection. Cyborg Bionic Syst. **5**, 0097 (2024)
43. Li, Z., Jia, J., Shi, Y.: MonoLSS: Learnable sample selection for monocular 3D detection. In: 3DV. pp. 1125–1135 (2024)
44. Li, Z., Chen, Z., Li, A., Fang, L., Jiang, Q., Liu, X., Jiang, J.: Unsupervised domain adaptation for monocular 3D object detection via self-training. In: ECCV. vol. 13669, pp. 245–262 (2022)
45. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: ECCV. vol. 13669, pp. 1–18 (2022)
46. Li, Z., Qu, Z., Zhou, Y., Liu, J., Wang, H., Jiang, L.: Diversity matters: Fully exploiting depth clues for reliable monocular 3D object detection. In: CVPR. pp. 2781–2790 (2022)
47. Lian, Q., Li, P., Chen, X.: MonoJSG: Joint semantic and geometric cost volume for monocular 3D object detection. In: CVPR. pp. 1060–1069 (2022)
48. Liu, H., Wu, C., Cheng, J., Chai, W., Wang, S., Liu, G., Hwang, J., Shuai, H., Cheng, W.: MonoTAKD: Teaching assistant knowledge distillation for monocular 3D object detection. In: CVPR. pp. 22266–22275 (2025)
49. Liu, X., Xue, N., Wu, T.: Learning auxiliary monocular contexts helps monocular 3D object detection. In: AAAI. pp. 1810–1818 (2022)
50. Liu, X., Zheng, C., Cheng, K., Xue, N., Qi, G., Wu, T.: Monocular 3D object detection with bounding box denoising in 3D by perceiver. In: ICCV. pp. 6413–6423 (2023)
51. Liu, Z., Zhou, D., Lu, F., Fang, J., Zhang, L.: AutoShape: Real-time shape-aware monocular 3D object detection. In: ICCV. pp. 15621–15630 (2021)
52. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: ICLR (2017)
53. Lu, Y., Ma, X., Yang, L., Zhang, T., Liu, Y., Chu, Q., Yan, J., Ouyang, W.: Geometry uncertainty projection network for monocular 3D object detection. In: ICCV. pp. 3091–3101 (2021)
54. Lv, W., Zhao, Y., Chang, Q., Huang, K., Wang, G., Liu, Y.: RT-DETRv2: Improved baseline with bag-of-freebies for real-time detection transformer. arXiv:2407.17140 [cs.CV] (2024)
55. Ma, X., Liu, S., Xia, Z., Zhang, H., Zeng, X., Ouyang, W.: Rethinking pseudo-LiDAR representation. In: ECCV. vol. 12358, pp. 311–327 (2020)
56. Ma, X., Zhang, Y., Xu, D., Zhou, D., Yi, S., Li, H., Ouyang, W.: Delving into localization errors for monocular 3D object detection. In: CVPR. pp. 4721–4730 (2021)
57. Mansourian, A.M., Ahmadi, R., Ghafouri, M., Babaei, A.M., Golezani, E.B., Ghamchi, Z.Y., Ramezanian, V., Taherian, A., Dinashi, K., Miri, A., Kasaei, S.: A comprehensive survey on knowledge distillation. Trans. Mach. Learn. Res. (2025)

58. Meier, J., Inchingolo, L., Dhaouadi, O., Xia, Y., Kaiser, J., Cremers, D.: MonoCT: Overcoming monocular 3D detection domain shift with consistent teacher models. In: ICRA. pp. 351–358 (2025)
59. Meier, J., Scalerandi, L., Dhaouadi, O., Kaiser, J., Nikita, A., Cremers, D.: CARLA Drone: Monocular 3D object detection from a different perspective. In: GCPR. pp. 137–152 (2024)
60. Pan, C., Peng, J., Zhang, Z.: Depth-guided vision transformer with normalizing flows for monocular 3D object detection. IEEE/CAA J. Autom. Sinica **11**(3), 673–689 (2024)
61. Peng, L., Xu, J., Cheng, H., Yang, Z., Wu, X., Qian, W., Wang, W., Wu, B., Cai, D.: Learning occupancy for monocular 3D object detection. In: CVPR. pp. 10281–10292 (2024)
62. Peng, Y., Li, H., Wu, P., Zhang, Y., Sun, X., Wu, F.: D-FINE: Redefine regression task in DETRs as fine-grained distribution refinement. In: ICLR (2025)
63. Pu, F., Wang, Y., Deng, J., Yang, W.: MonoDGP: Monocular 3D object detection with decoupled-query and geometry-error priors. In: CVPR. pp. 6520–6530 (2025)
64. Ranasinghe, Y., Hegde, D., Patel, V.M.: MonoDiff: Monocular 3D object detection and pose estimation with diffusion models. In: CVPR. pp. 10659–10670 (2024)
65. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distribution network for monocular 3D object detection. In: CVPR. pp. 8555–8564 (2021)
66. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NeurIPS. vol. 28, pp. 91–99 (2015)
67. Rezaei, M., Azarmi, M., Mir, F.M.P.: 3D-Net: Monocular 3D object recognition for traffic monitoring. Expert Syst. Appl. **227**, 120253 (2023)
68. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs.CL] (2019)
69. Shi, H., Pang, C., Zhang, J., Yang, K., Wu, Y., Ni, H., Lin, Y., Stiefelhagen, R., Wang, K.: CoBEV: Elevating roadside 3D object detection with depth and height complementarity. IEEE Trans. Image Process. **33**, 5424–5439 (2024)
70. Shi, X., Chen, Z., Kim, T.: Multivariate probabilistic monocular 3D object detection. In: WACV. pp. 4270–4279 (2023)
71. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., et al.: Scalability in perception for autonomous driving: Waymo Open Dataset. In: CVPR. pp. 2443–2451 (2020)
72. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: ICML. vol. 70, pp. 3319–3328 (2017)
73. Tang, Z., Naphade, M., Liu, M., Yang, X., Birchfield, S., Wang, S., Kumar, R., Anastasiu, D.C., Hwang, J.: CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In: CVPR. pp. 8797–8806 (2019)
74. Tian, Y., Ye, Q., Doermann, D.S.: YOLOv12: Attention-centric real-time object detectors. arXiv:2502.12524 [cs.CV] (2025)
75. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G.: YOLOv10: Real-time end-to-end object detection. In: NeurIPS. vol. 37, pp. 107984–108011 (2024)
76. Wang, L., Zhang, L., Zhu, Y., Zhang, Z., He, T., Li, M., Xue, X.: Progressive coordinate transforms for monocular 3D object detection. In: NeurIPS. vol. 34, pp. 13364–13377 (2021)

77. Wang, S., Xia, C., Lv, F., Shi, Y.: RT-DETRv3: Real-time end-to-end object detection with hierarchical dense positive supervision. In: WACV. pp. 1628–1636 (2025)
78. Wu, Z., Gan, Y., Wu, Y., Wang, R., Wang, X., Pu, J.: FD3D: Exploiting foreground depth map for feature-supervised monocular 3D object detection. In: AAAI. pp. 6189–6197 (2024)
79. Wu, Z., Song, F., Gan, Y., Wu, Y., Xu, T., Wang, X., Tang, R., Pu, J.: Advancing 3D object detection with depth-aware spatial knowledge distillation. IEEE Trans. Pattern Anal. Mach. Intell. **47**(11), 10295–10310 (2025)
80. Wu, Z., Wu, Y., Pu, J., Li, X., Wang, X.: Attention-based depth distillation with 3D-aware positional encoding for monocular 3D object detection. In: AAAI. pp. 2892–2900 (2023)
81. Xu, J., Peng, L., Chen, H., Li, H., Qian, W., Li, K., Wang, W., Cai, D.: MonoNeRD: NeRF-like representations for monocular 3D object detection. In: CVPR. pp. 6791–6801 (2023)
82. Yan, L., Yan, P., Xiong, S., Xiang, X., Tan, Y.: MonoCD: Monocular 3D object detection with complementary depths. In: CVPR. pp. 10248–10257 (2024)
83. Yan, W., Xu, L., Liu, H., Tang, C., Zhou, W.: High-order structural relation distillation networks from LiDAR to monocular image 3D detectors. IEEE Trans. Intell. Veh. **9**(2), 3593–3604 (2024)
84. Yang, F., Chen, H., He, Y., Zhao, S., Zhang, C., Ni, K., Ding, G.: Geometry-guided domain generalization for monocular 3D object detection. In: AAAI. pp. 6467–6476 (2024)
85. Yang, F., He, X., Chen, W., Zhou, P., Li, Z.: MonoPSTR: Monocular 3D object detection with dynamic position and scale-aware transformer. IEEE Trans. Instrum. Meas. **73**, 1–13 (2024)
86. Yang, L., Yu, K., Tang, T., Li, J., Yuan, K., Wang, L., Zhang, X., Chen, P.: BEVHeight: A robust framework for vision-based roadside 3D object detection. In: CVPR. pp. 21611–21620 (2023)
87. Yang, Y.H., Piccinelli, L., Segu, M., Li, S., Huang, R., Fu, Y., Pollefeys, M., Blum, H., Bauer, Z.: 3D-MOOD: Lifting 2D to 3D for monocular open-set object detection. In: ICCV. pp. 7429–7439 (2025)
88. Ye, X., Shu, M., Li, H., Shi, Y., Li, Y., Wang, G., Tan, X., Ding, E.: Rope3D: The roadside perception dataset for autonomous driving and monocular 3D object detection task. In: CVPR. pp. 21309–21318 (2022)
89. Yun, S., Han, D., Chun, S., Oh, S.J., Yoo, Y., Choe, J.: CutMix: Regularization strategy to train strong classifiers with localizable features. In: ICCV. pp. 6022–6031 (2019)
90. Zhang, H., Cissé, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: ICLR (2018)
91. Zhang, J., Li, J., Lin, X., Zhang, W., Tan, X., Han, J., Ding, E., Wang, J., Li, G.: Decoupled pseudo-labeling for semi-supervised monocular 3D object detection. In: CVPR. pp. 16923–16932 (2024)
92. Zhang, R., Qiu, H., Wang, T., Guo, Z., Cui, Z., Qiao, Y., Li, H., Gao, P.: MonoDETR: Depth-guided transformer for monocular 3D object detection. In: CVPR. pp. 9121–9132 (2023)
93. Zhang, Y., Lu, J., Zhou, J.: Objects are different: Flexible monocular 3D object detection. In: CVPR. pp. 3289–3298 (2021)
94. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., Chen, J.: DETRs beat YOLOs on real-time object detection. In: CVPR. pp. 16965–16974 (2024)

95. Zhenyu, L., Zehui, C., Ang, L., Liangji, F., Jiang, Q., Xianming, L., Junjun, J.: Towards model generalization for monocular 3D object detection. arXiv:2205.11664 [cs.CV] (2022)
96. Zhou, Y., Liu, Q., Zhu, H., Li, Y., Chang, S., Guo, M.: Exploiting ground depth estimation for mobile monocular 3D object detection. IEEE Trans. Pattern Anal. Mach. Intell. **47**(4) (2025)
97. Zhu, X., Sheng, H., Cai, S., Deng, B., Yang, S., Liang, Q., Chen, K., Gao, L., Song, J., Ye, J.: RoScenes: A large-scale multi-view 3D dataset for roadside perception. In: ECCV. vol. 15099, pp. 331–347 (2024)