Rethinking Geometric Depth in Monocular 3D Object Detection: A Projection-Consistent Reformulation

Zhihao Zhang[1] **Abhinav Kumar**[1] **Huaizhi Qu**[2] **Tianlong Chen**[2] **Xiaoming Liu**[1,2]
[1]Michigan State University [2]University of North Carolina at Chapel Hill

Abstract

Monocular 3D detection (Mono3D) hinges on recovering depth from a single image, for which the geometric formulation $z \approx f \cdot H/h_{2D}$ is widely adopted due to its simplicity and direct grounding in projective geometry. We show that this formulation harbors a systematic, projection-induced bias: substituting the 2D bounding-box height h_{2D} recovers the depth of an object's visible front surface rather than its center, causing size-dependent underestimation that persists even under ground-truth inputs. We address this front-surface bias with MonoPCA, a unified framework built on a projection-consistent reformulation. At its core, PCDepth retains the measurable h_{2D} and closes the gap to center depth through an orientation- and size-aware correction derived from perspective geometry, eliminating the bias without introducing ill-posed regression targets. Because PCDepth couples depth to predicted object size and orientation, attribute perception becomes essential. We therefore introduce GeoAlign, a teacher–student framework that transfers geometric representations from Vision Foundation Models into a lightweight student, strengthening attribute perception at no inference cost. Across four challenging benchmarks, MonoPCA achieves SoTA performance, including a $+1.44\%$ AP_{3D} gain on the KITTI leaderboard and $+2.80\%$ on $Omni3D_{OUT}$. Code is available at https://github.com/alanzhangcs/MonoPCA.

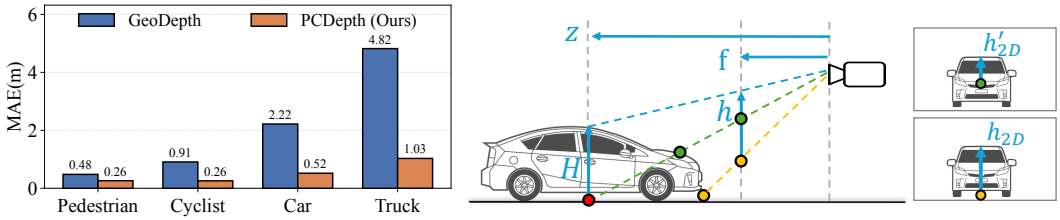
1 Introduction

Monocular 3D object detection (Mono3D) seeks to recover object location, size, and orientation from a single RGB image, underpinning applications from autonomous driving [71, 49, 52] to augmented reality [54, 58]. Yet a single image fundamentally lacks metric scale: projecting a 3D scene onto a 2D plane collapses absolute distance into a one-to-many ambiguity. Depth estimation thus emerges as an ill-posed inverse problem and, in practice, the main bottleneck of Mono3D [49].

A substantial body of work has sought to tame this ambiguity, broadly along two complementary axes. The first injects geometric and probabilistic priors into depth prediction, including keypoint-based reasoning in MonoFlex [90], geometric decomposition in MonoRCNN [69], and uncertainty modeling in GUP Net [47]. The second reshapes the depth representation itself, through scale-equivariant features in DEVIANT [31], depth-guided decoding in MonoDETR [89], complementary depth heads in MonoCD [83], decoupled object queries in MonoDGP [62], and inter-attribute coupling in MonoCoP [94]. Despite their architectural diversity, these methods rest on a shared foundation: when grounded geometrically, depth is recovered through the projection relation $z \approx f \cdot H/h$, with h instantiated as the predicted 2D bounding-box height h_{2D} [69, 47, 31, 89, 62, 94]. This shared foundation, we find, harbors a systematic bias that has gone unexamined.

We observe that the formula $z = f \cdot H/h_{2D}$ exhibits a persistent depth error even under *ground-truth* inputs (Fig. 1a), and this error grows monotonically with object size, an unmistakable signature of a structural rather than statistical flaw. Tracing the formula back to its geometric origin reveals the cause. Under perspective projection, h_{2D} does not encode the depth of the object center but that

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Depth Error (b) 3D-to-2D Perspective Projection

Figure 1: **(a) Depth error under ideal inputs.** We evaluate the GeoDepth formulation [69] using ground-truth f, H, and h_{2D}. Even with these oracle inputs, GeoDepth incurs a non-trivial error that grows with object scale, revealing a structural rather than statistical bias. Our PCDepth substantially mitigates this bias across all categories. **(b) Geometric origin of the bias.** Under perspective projection, the 3D bottom center (red) is self-occluded by a nearer surface point (green) that projects higher on the image plane, while the visible bottom in the image corresponds to a still-closer surface point (yellow). The commonly adopted formulation $z = f \cdot H/h_{2D}$ therefore recovers the depth of the visible front surface rather than that of the object center, systematically underestimating depth.

of the visible front surface. For any object with non-negligible thickness, the 3D bottom center is self-occluded by a nearer surface point that projects higher on the image plane (Fig. 1b). The effective projective height tied to the center, h'_{2D}, therefore satisfies $h'_{2D} < h_{2D}$, and substituting h_{2D} into the projection relation systematically underestimates center depth. The bias is small for thin objects and severe for large vehicles, precisely the pattern observed in Fig. 1a.

A naive remedy would be to regress h'_{2D} directly, but h'_{2D} is depth-dependent and not directly observable, making its regression circular and unstable. We instead retain the measurable h_{2D} and close the gap analytically. Casting the discrepancy as the object's half-extent along the optical axis, we derive an orientation- and size-aware correction expressible in closed form. The result is PCDepth, a projection-consistent depth formulation that recovers center depth from h_{2D}, the 3D size, and the yaw angle, eliminating the front-surface bias of geometric estimation (Fig. 1a).

A consequence of PCDepth is that depth is no longer an isolated scalar but a function of the predicted 3D size, orientation, and 2D height. This coupling is a double-edged sword: it removes the projection bias but propagates attribute errors into depth, making accurate attribute perception essential. Lightweight detectors typical of Mono3D often underfit the global semantic context required to estimate size and orientation reliably, while simply scaling up the backbone forfeits inference efficiency. We resolve this tension with GeoAlign, a teacher–student framework in which a high-capacity teacher is built via a Mixture-of-Backbone (MoB) that fuses a convolutional hierarchy with Vision Foundation Models (VFMs), and is then distilled into a standard ResNet student through Hierarchical Feature Alignment (HFA). The student inherits the teacher's geometric perception at no additional inference cost, fully exploiting the potential of PCDepth.

In summary, our main contributions are as follows:

• We identify a systematic, projection-induced bias in the conventional geometric depth formulation: substituting the 2D bounding-box height h_{2D} recovers the depth of the visible front surface rather than the object center, causing size-dependent depth underestimation overlooked by prior work.

• We propose MonoPCA, a unified framework comprising (1) PCDepth, a projection-consistent depth formulation with orientation- and size-aware corrections that eliminates the projection bias, and (2) GeoAlign, a teacher–student distillation framework that transfers VFM-enriched geometric representations to a lightweight student without inference overhead.

• Extensive experiments demonstrate that MonoPCA achieves SoTA performance on four challenging benchmarks, including a $+1.44\%$ AP_{3D} gain on the KITTI leaderboard and $+2.80\%$ on $Omni3D_{OUT}$, with consistent improvements across object categories and distance ranges.

2 Related Work

Mono3D. Existing Mono3D methods fall into two architectural paradigms. *(1) CNN-based methods* [67] extend 2D detectors to 3D using geometric priors and structural constraints [4, 32, 90,

45, 51, 5], with center-based pipelines [45, 51, 78, 44] predicting 3D boxes from image centers and others modeling 2D–3D correlations [38, 47, 90, 82]. Complementary lines explore scale-equivariant, camera-aware, and weather-robust designs [31, 64, 95, 34, 23, 17, 55], multi-scale FPN structures [4, 32, 3], BEV segmentation for large objects [33], and auxiliary supervision from CAD models [9, 46, 35, 30, 41], LiDAR [50, 79, 10, 66, 61, 15, 19, 22, 42], synthetic scenes [39, 40, 20, 57], and dense depth maps [16, 63, 25, 48, 60, 65], as well as label-efficient training with weak, sparse, or pseudo labels [87, 27, 73, 36, 53]; we refer readers to [49] for a comprehensive survey. *(2) Transformer-based methods* [8, 11, 97] bring end-to-end detection to Mono3D [21, 81, 98, 89, 62, 80, 59, 28], with MonoDETR [89] introducing depth-guided queries, MonoDGP [62] decoupling visual decoding from error-based depth, and MonoCoP [94] modeling inter-attribute correlations. MonoPCA is orthogonal to these architectural advances: rather than redesigning the detector, it targets the depth formulation that all of them share, and can be plugged into both CNN- and transformer-based detectors as a drop-in replacement for the depth head, as we demonstrate in Fig. 7.

Geometric Depth Estimation in Mono3D. Geometric projection has long been used to compensate for the lack of metric scale in monocular images, relating observable 2D cues to 3D distance through camera intrinsics [1, 69, 2, 7, 47, 89, 62, 84]. Across these methods, the projection relation $z = fH/h_{2D}$ is adopted as a given, with effort directed at the surrounding pipeline (*e.g.*, keypoint reasoning [90], geometric priors [69], or uncertainty modeling [47]) rather than the formula itself. While related lines of work adopt a surface-plus-offset decomposition for regression-based depth [60, 29] or refine predicted depth post hoc [96], the analytical formula and its bias remain unaddressed. We instead identify that substituting the 2D bounding-box height for the projected center height introduces a systematic projection bias, a structural flaw of the formula that has gone unexamined despite its broad use [47, 89, 62, 94], and derive an orientation- and size-aware correction directly from perspective geometry.

Knowledge Distillation in Mono3D. Distillation is a common way to inject depth knowledge into monocular detectors, typically from LiDAR-based teachers whose inputs are unavailable at inference [15, 42]. GeoAlign differs in that its teacher consumes the same monocular image as the student, and it transfers VFM-enriched features for attribute perception rather than depth.

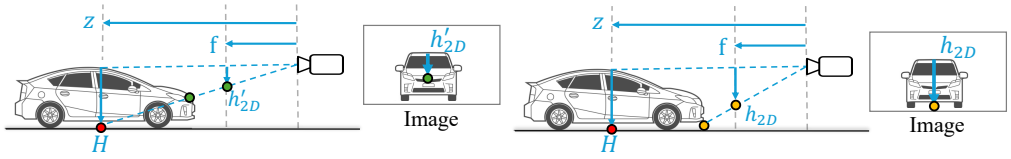
Vision Foundation Models. Vision Foundation Models (VFMs) learn transferable representations from large-scale vision–language or self-supervised pretraining, with representative families including CLIP [76, 86] and DINO [56, 70]. While these backbones provide strong semantic priors and have been adapted to 3D shape understanding [91], their use in geometry-sensitive tasks such as Mono3D remains limited [95, 96, 85]. We leverage VFMs in a teacher–student framework to construct enriched multi-scale representations that support projection-consistent depth modeling under PCDepth.

3 Method

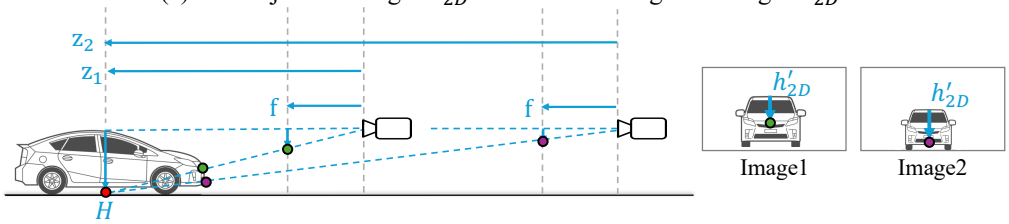
3.1 Problem Setup and Overview

Task. Given a single RGB image and known camera intrinsics, Mono3D aims to detect every object as a 3D bounding box parameterized by its center (x, y, z), physical size (W, H, L), and yaw angle γ in the camera coordinate frame, alongside a class label C and a 2D bounding box on the image plane. Among these unknowns, the depth z is the dominant source of error: a small relative error in z produces a metric displacement that scales with depth, making it the central bottleneck of Mono3D.

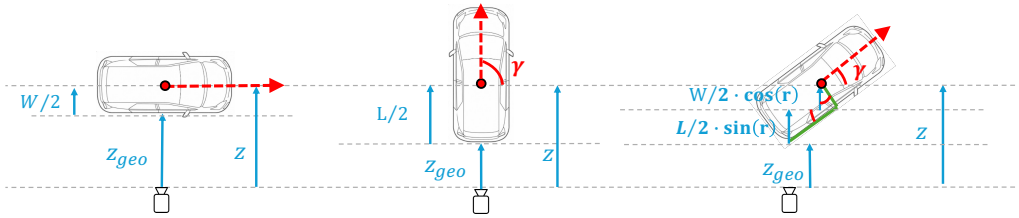
Approach at a glance. We pursue accurate depth from two angles (see Fig. 3). First, we revisit the geometric depth formulation that underpins most modern detectors and identify a structural bias overlooked by prior work (Sec. 3.2). Second, we correct this bias with PCDepth, a projection-consistent depth formulation that derives center depth from observable 2D evidence together with predicted 3D attributes through geometric correction (Sec. 3.3). Because PCDepth couples depth to the predicted size and orientation, depth accuracy now hinges on attribute accuracy. We close this loop with GeoAlign (Sec. 3.4), which distills geometry-aware representations from Vision Foundation Models into a lightweight student, strengthening attribute perception at no inference cost.



(a) 2D Projective Height h'_{2D} vs. 2D Bounding Box Height h_{2D}



(b) 2D Projective Height h'_{2D} is Depth-Dependent



(c) Geometric interpretation of PCDepth

Figure 2: Geometric basis of PCDepth. **(a) Source of the front-surface bias.** Under perspective projection, the 3D bottom center (red) is self-occluded by a nearer surface point (green) that projects higher on the image plane, while the visible bottom of the box corresponds to a still-closer point (yellow), giving $h'_{2D} < h_{2D}$. The conventional estimate $z = f \cdot H/h_{2D}$ therefore recovers the front-surface depth rather than the center depth. **(b) h'_{2D} is depth-dependent.** As the center depth changes from z_1 to z_2, the surface point defining h'_{2D} shifts (green to purple), so h'_{2D} is not directly observable and its regression is ill-posed. **(c) Recovering center depth via an optical-axis correction.** z_{geo} recovers the front-surface depth, displaced from the object center by an offset Δ along the optical axis: $\Delta \approx W/2$ at $\gamma = 0°$ (**Left**), $\Delta \approx L/2$ at $\gamma = 90°$ (**Middle**), and $\Delta \approx \frac{L}{2}|\sin\gamma| + \frac{W}{2}|\cos\gamma|$ for arbitrary yaw (**Right**), giving $z = z_{\text{geo}} + \Delta$.

3.2 The Front-Surface Bias of Geometric Depth

The geometric depth formulation traces back to the pinhole camera model, under which the depth of a point at 3D height H projecting to image height h is

$$z = f \cdot \frac{H}{h}, \tag{1}$$

with f the focal length. In the Mono3D literature [89, 47, 62], h is instantiated by the predicted 2D bounding-box height h_{2D}, yielding $z_{\text{geo}} = f \cdot H/h_{2D}$. This substitution is intuitive but, as we now show (Fig. 2a), geometrically inconsistent.

Diagnosis. Eq. (1) is exact only when h is the projection of the line segment passing through the object center perpendicular to the optical axis. For an object with non-negligible thickness, this center-aligned segment is *not* what the 2D bounding box captures. As shown in Fig. 2a, the 3D bottom center (red) is self-occluded by a nearer surface point (green) that projects higher on the image plane; the visible bottom of the box corresponds to a still-closer surface point (yellow). The effective center-aligned projective height h'_{2D} therefore satisfies $h'_{2D} < h_{2D}$, and z_{geo} recovers the depth of the visible front surface rather than that of the object center. We refer to the resulting offset as the *front-surface bias*; Fig. 1a shows that it grows monotonically with object size and persists even under ground-truth f, H, and h_{2D}, confirming a structural rather than statistical origin.

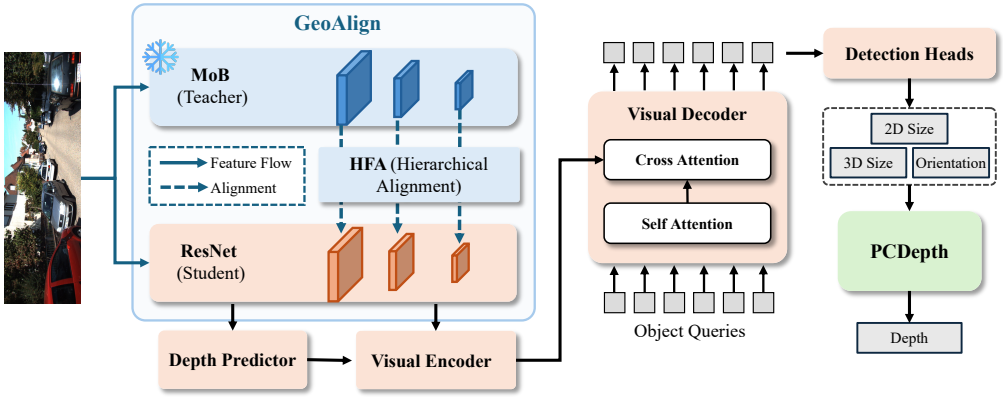


Figure 3: Overview of MonoPCA. The conventional geometric depth $z = f \cdot H/h_{2D}$ recovers the visible front surface rather than the object center, causing a systematic projection bias. PCDepth corrects this bias analytically by adding an orientation- and size-aware offset, making depth an explicit function of the predicted 3D size and yaw. This shifts the bottleneck onto attribute accuracy, which GeoAlign strengthens by distilling representations from a Mixture-of-Backbone (MoB) teacher into a ResNet student via Hierarchical Feature Alignment (HFA).

3.3 PCDepth: A Projection-Consistent Reformulation

Why not regress $\mathbf{h'_{2D}}$? The most direct fix is to predict h'_{2D} and substitute it into Eq. (1). This is, however, ill-posed: h'_{2D} is not a property that can be read off the image but is itself a function of the depth we seek,

$$h'_{2D} = g(z;\, \gamma, \mathbf{s}), \tag{2}$$

where $\mathbf{s} = (W, H, L)$ is the 3D size. Fig. 2b illustrates this dependency: changing z shifts the visible surface point that defines h'_{2D}, so any regressor for h'_{2D} is implicitly conditioned on the very depth it is meant to estimate, creating a circular dependency that destabilizes optimization.

What does $\mathbf{h_{2D}}$ actually encode? We instead retain the directly measurable h_{2D} and derive its geometric meaning from first principles. Under the pinhole model, the eight corners of the 3D bounding box project to image points $\big(fX_i/Z_i,\ fY_i/Z_i\big)$, and h_{2D} is the spread of their vertical projections. For an upright box with yaw-only orientation, as in standard driving benchmarks, all four top corners share $Y = -H/2$ and all four bottom corners share $Y = +H/2$, so the extremal image-plane y-values are both attained at the corner with the *minimum* depth $Z_{\min}$. This gives

$$h_{2D} \;=\; \frac{fH}{Z_{\min}}, \qquad Z_{\min} \;=\; z - \Delta, \tag{3}$$

where z is the center depth and Δ is the optical-axis half-extent of the box footprint. Substituting into the conventional formula makes the front-surface bias explicit:

$$z_{\text{geo}} \;=\; f \cdot \frac{H}{h_{2D}} \;=\; Z_{\min} \;=\; z - \Delta, \qquad \text{i.e.,} \qquad z \;=\; z_{\text{geo}} + \Delta. \tag{4}$$

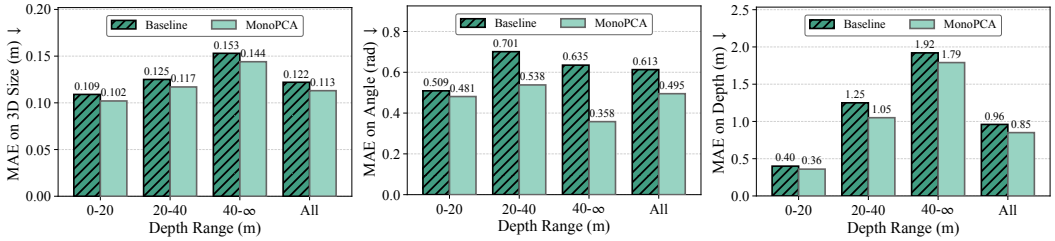
z_{geo} recovers the depth of the box's nearest corner along the optical axis rather than that of the center, and the gap is exactly Δ.

Closed-form correction. For a box yawed by γ, the four bottom-footprint corners have depths $z \pm (L/2)\sin\gamma \pm (W/2)\cos\gamma$, so the minimum is attained at the corner closest to the camera:

$$\Delta \;=\; \frac{L}{2}|\sin\gamma| + \frac{W}{2}|\cos\gamma|, \tag{5}$$

which reduces to $\Delta = W/2$ at $\gamma = 0°$ (object facing the camera) and $\Delta = L/2$ at $\gamma = 90°$ (object sideways), matching the limit cases in Fig. 2c. Combining Eq. (4) and Eq. (5) yields PCDepth:

$$z \;=\; f \cdot \frac{H}{h_{2D}} \;+\; \frac{L}{2}|\sin\gamma| \;+\; \frac{W}{2}|\cos\gamma|. \tag{6}$$



(a) 3D Size Error (b) Orientation Error (c) Depth Error

Figure 4: GeoAlign reduces depth error through the attributes PCDepth depends on. Per-distance errors on KITTI Val for 3D size **(Left)**, orientation **(Middle)**, and depth **(Right)**. GeoAlign consistently lowers size and orientation errors across all ranges, and since PCDepth computes depth analytically from these two attributes, the improvements propagate directly to depth.

Under pinhole projection and an upright-box orientation, this identity holds analytically when h_{2D} is the tight projection of the eight 3D corners. In real datasets, h_{2D} may deviate from this tight projection due to truncation, occlusion, or annotation noise, but PCDepth still provides a stronger geometric guarantee than regression-based formulations, as confirmed in Fig. 1a.

3.4 GeoAlign: Strengthening Attribute Perception

PCDepth ties depth accuracy to the predicted size and yaw, shifting the bottleneck to attribute perception. Both attributes are difficult to recover from a single image: size requires category-level priors inferred without metric scale, and yaw relies on subtle appearance cues such as part configuration and self-shading. Such cues call for representations that combine fine local geometry with long-range semantic context, a regime where Vision Foundation Models (VFMs) excel, yet naive backbone replacement would multiply inference cost. GeoAlign resolves this tension by distilling a high-capacity teacher, formed by fusing a VFM with the convolutional backbone, into a lightweight student, transferring attribute perception (Fig. 4) without inference overhead.

Mixture-of-Backbone. The two branches are structurally incompatible out of the box: the convolutional backbone produces a four-level pyramid at strides $\{1/8, 1/16, 1/32, 1/64\}$, with the last level obtained by a stride-2 convolution on the $1/32$ map, while the VFM emits a single-scale representation. MoB reconciles them in two steps (Fig. 5).

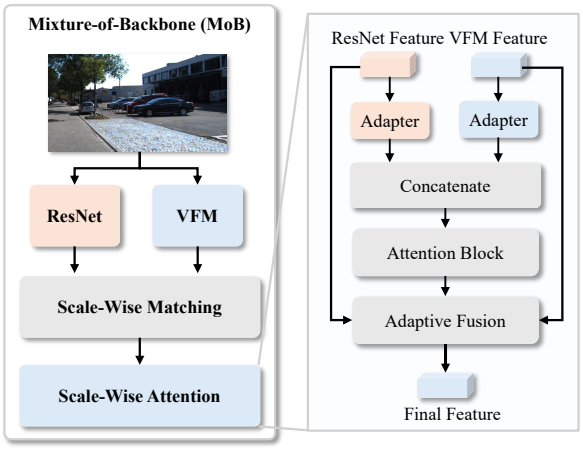


Figure 5: Details of Mixture-of-Backbone. MoB adaptively fuses a convolutional pyramid with a single-scale VFM branch at every scale.

Scale-Wise Matching expands the VFM output into a four-level pyramid through an FPN [37], producing F_s^V at each scale s spatially aligned with the convolutional features F_s^R.

Scale-Wise Attention fuses the aligned features adaptively. Lightweight adapters ϕ_R, ϕ_V map both streams into a shared embedding space, $\tilde{F}_s^R = \phi_R(F_s^R)$ and $\tilde{F}_s^V = \phi_V(F_s^V)$. A small attention module then produces a per-location weight map and the fused teacher feature:

$$A_s = \sigma\big(\mathcal{A}([\tilde{F}_s^R, \tilde{F}_s^V])\big), \qquad F_s^T = A_s \odot \tilde{F}_s^R + (1 - A_s) \odot \tilde{F}_s^V, \tag{7}$$

with σ the element-wise sigmoid. By learning A_s per location, MoB defers to convolutional features where local geometric detail dominates and to the VFM where semantic context dominates.

Hierarchical Feature Alignment. Given the teacher features $\{F_s^T\}_{s=1}^4$ and student features $\{F_s^S\}_{s=1}^4$ from a standard ResNet, HFA aligns the two patch-wise across all scales. We project

Table 1: KITTI Leaderboard (Test) and Val results at $\text{IoU}_{3D} \geq 0.7$. Without extra data, MonoPCA achieves SoTA performance across all metrics on both the KITTI leaderboard (Test) and KITTI val. [Key: **Best**, Second Best, **Gain**]

Method	Extra Data	**Test**, AP_{3D} ↑			**Test**, AP_{BEV} ↑			**Val**, AP_{3D} ↑			**Val**, AP_{BEV} ↑		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
OccupancyM3D [61]	LiDAR	25.55	17.02	14.79	35.38	24.18	21.37	26.87	19.96	17.15	35.72	26.60	23.68
OPA-3D [74]	Depth	24.68	17.17	14.14	32.50	23.14	20.30	24.97	19.40	16.59	33.80	25.51	22.13
MonoTAKD [42]	LiDAR	27.91	19.43	16.51	38.75	27.76	24.14	34.36	22.61	19.88	42.86	29.41	26.47
MonoCon [43]	None	22.50	16.46	13.95	31.12	22.10	19.00	26.33	19.01	15.98	–	–	–
MonoDETR [89]		25.00	16.47	13.58	33.60	22.11	18.60	28.84	20.61	16.38	37.86	26.95	22.80
MonoUNI [24]		24.75	16.73	13.49	–	–	–	24.51	17.18	14.01	–	–	–
FD3D [82]		25.38	17.12	14.50	34.20	23.72	20.76	28.22	20.23	17.04	36.98	26.77	23.16
MonoMAE [26]		25.60	18.84	16.78	34.15	24.93	21.76	30.29	20.90	17.61	40.26	27.08	23.14
MonoCD [83]		25.53	16.59	14.53	33.41	22.81	19.57	26.45	19.37	16.38	34.60	24.96	21.51
MonoDGP [62]		26.35	18.72	15.97	35.24	25.23	22.02	30.76	22.34	19.02	39.40	28.20	24.42
MonoCoP [94]		27.54	19.11	16.33	36.77	25.57	22.62	32.06	23.98	20.64	41.92	30.75	27.06
MonoPCA		**30.34**	**20.55**	**17.84**	**38.65**	**26.63**	**24.43**	**35.86**	**26.22**	**23.58**	**46.29**	**33.89**	**30.05**
Δ		**+2.80**	**+1.44**	**+1.06**	**+1.88**	**+1.06**	**+1.81**	**+3.80**	**+2.24**	**+2.94**	**+4.37**	**+3.14**	**+2.99**

the student into the teacher's embedding space via $\hat{F}_s^S = \text{Proj}(F_s^S)$ and minimize the patch-wise cosine distance:

$$\mathcal{L}_{\text{align}} = \sum_{s=1}^{4} \frac{1}{N_s} \sum_{i=1}^{N_s} \left(1 - \frac{\langle \hat{F}_{s,i}^S, F_{s,i}^T \rangle}{\|\hat{F}_{s,i}^S\|_2 \, \|F_{s,i}^T\|_2} \right), \tag{8}$$

where i indexes spatial locations and N_s the number of patches at scale s. Cosine is invariant to the VFM-induced magnitude gap between teacher and student, and we span all four scales to capture attribute cues at multiple resolutions.

Overall Loss. The training objective combines the standard DETR-style detection losses with the alignment term, $\mathcal{L} = \mathcal{L}_{\text{det}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}}$, where $\mathcal{L}_{\text{det}}$ supervises Hungarian-matched predictions with classification, 2D box, projected-center, depth, size, orientation, and depth-map losses, and λ_{align} balances the two terms. The individual losses and their weights are given in Appendix A.

4 Experiments

Datasets. We evaluate MonoPCA on four representative Mono3D benchmarks. **KITTI** [18] contains 7,481 training and 7,518 testing images; following [12], the training split is further divided into 3,712 training and 3,769 validation images. **nuScenes** [6] provides 28,130 training and 6,019 validation front-camera images, with annotations converted to KITTI format following [31]. **Waymo** [75] provides 52,386 training and 39,848 validation front-view samples [31]. **Omni3D** [3] aggregates multiple 3D detection datasets into a unified benchmark; we evaluate on $\text{Omni3D}_{\text{OUT}}$, $\text{Omni3D}_{\text{KITTI}}$, and $\text{Omni3D}_{\text{nuScenes}}$, and additionally report $\text{Omni3D}_{\text{OUT}}^{\text{Small}}$, $\text{Omni3D}_{\text{OUT}}^{\text{Large}}$, and $\text{Omni3D}_{\text{OUT}}^{\text{Car}}$ splits to disentangle performance across object scales.

Evaluation Metrics. For KITTI and nuScenes, we follow [72] and report Car AP_{3D} and AP_{BEV} at $\text{IoU}_{3D} \geq 0.7$, with Moderate as the primary setting, and additionally report $\text{IoU}_{3D} \geq 0.5$ on nuScenes. For Waymo, we report AP_{3D} and the heading-aware APH_{3D} [66, 31] at $\text{IoU}_{3D} \geq 0.5$ and 0.7 over distance ranges $[0, 30)$, $[30, 50)$, and $[50, \infty)$ meters. For Omni3D, we report AP averaged over IoU_{3D} thresholds $\tau \in [0.05, 0.10, \ldots, 0.50]$.

Implementation Details. We build MonoPCA on MonoCoP [94]. The student uses a ResNet-50 backbone, while the teacher fuses ResNet-50 with EVA-CLIP [76] through MoB and is discarded after training. We use AdamW with an initial learning rate of 2×10^{-4} (2×10^{-5} for the teacher backbone) and a weight decay of 10^{-4}, and halve the learning rate at epochs 85, 125, 165, and 205. The alignment loss weight is $\lambda_{align} = 5$ (Appendix G.4). All models are trained with a batch size of 16 on eight NVIDIA A100 GPUs; a full KITTI run takes about 12 hours. The full architecture, training losses, and loss weights are given in Appendix A.

Table 2: nuScenes Val results. MonoPCA achieves SoTA performance on both 3D and BEV detection under IoU thresholds 0.7 and 0.5. [Key: **Best**, Second Best, **Gain**]

Method	$\mathbf{IoU_{3D} \geq 0.7}$				$\mathbf{IoU_{3D} \geq 0.5}$			
	$AP_{3D} \uparrow$		$AP_{BEV} \uparrow$		$AP_{3D} \uparrow$		$AP_{BEV} \uparrow$	
	Easy	Mod.	Easy	Mod.	Easy	Mod.	Easy	Mod.
GUP Net [47]	8.50	7.40	14.21	12.81	29.03	26.16	33.42	30.23
DEVIANT [31]	9.69	8.33	16.28	14.36	31.47	28.22	35.61	31.93
MonoDETR [89]	9.53	8.19	16.39	14.41	31.81	28.35	35.70	31.96
MonoDGP [62]	10.04	8.78	16.55	14.53	29.56	26.17	32.67	29.44
MonoCoP [94]	10.85	9.71	17.83	15.86	33.70	29.91	37.44	34.01
MonoPCA	**13.75**	**12.03**	**20.97**	**18.49**	**38.50**	**34.11**	**42.03**	**38.49**
Δ	**+2.90**	**+2.32**	**+3.14**	**+2.63**	**+4.80**	**+4.20**	**+4.59**	**+4.48**

4.1 Standard 3D Benchmarks

We evaluate MonoPCA on three standard Mono3D benchmarks (KITTI [18], nuScenes [6], and Waymo [75]), where each model is trained and tested on a single dataset.

KITTI Leaderboard (Test) Results. The KITTI leaderboard withholds ground-truth annotations and evaluates submissions on the official server, providing a fully blind evaluation protocol. As shown in Tab. 1, MonoPCA achieves new SoTA performance across all 3D and BEV metrics without any extra supervision. Under the primary AP_{3D} Mod. metric, MonoPCA reaches 20.55%, outperforming MonoCoP by $+1.44\%$. The improvement holds across all difficulty levels and extends to BEV evaluation, indicating more accurate 3D localization. MonoPCA also surpasses methods that rely on additional supervision such as LiDAR or depth maps (*e.g.*, MonoTAKD and OPA-3D) on all AP_{3D} metrics. More results are provided in Appendix B.

KITTI Val Results. We observe similar trends on the Val split. MonoPCA achieves 26.22% under AP_{3D} Mod., exceeding MonoCoP by $+2.24\%$. The gains hold across all difficulty levels, including the Hard setting dominated by occluded and distant objects, suggesting that they stem from better geometric depth rather than from a subset of easy cases.

nuScenes Val Results. As shown in Tab. 2, the gains carry over to nuScenes: MonoPCA leads on every 3D and BEV metric and improves AP_{3D} Mod. by $+2.32\%$ over MonoCoP at IoU ≥ 0.7. The margin widens at IoU ≥ 0.5 ($+4.20\%$), suggesting that PCDepth removes a systematic depth offset that pushes predictions out of the matching range, rather than only tightening well-localized boxes.

Waymo Val Results. On Waymo (Appendix D), MonoPCA achieves the best overall AP_{3D} and APH_{3D} at IoU ≥ 0.5, with gains concentrated within 50 m ($+4.77\%$ and $+1.61\%$ AP_{3D} over MonoCoP at 0–30 m and 30–50 m). At these distances depth noise is small and the front-surface offset is a dominant error source, so correcting it pays off most; farther away, noise that grows with depth limits the benefit of any correction.

Efficiency Analysis. Beyond accuracy, Tab. 3 shows that MonoPCA is also more efficient on the KITTI leaderboard: AP_{3D} improves from 19.11% (MonoCoP) to 20.55%, while GFLOPs drop from 71.8 to 62.5 at comparable parameters. The GFLOPs reduction reflects an architectural simplification enabled by PCDepth: computing depth in closed form from already-predicted attributes removes the need for a dedicated depth encoder and depth-guided decoder. PCDepth is parameter-free and the GeoAlign teacher is discarded after training, so neither adds inference cost.

Table 3: Efficiency comparison.

Method	$AP_{3D} \uparrow$	Param(M) $\downarrow$	GFLOPs(G) $\downarrow$
MonoDGP [62]	18.72	38.90	69.0
MonoCoP [94]	19.11	42.50	71.8
MonoPCA	**20.55**	42.00	62.5

4.2 Omni3D Benchmark

Beyond single-dataset evaluations, we assess MonoPCA on the large-scale Omni3D benchmark, which jointly trains and evaluates on multiple datasets. This setting requires consistent performance across domains rather than overfitting to a single dataset. As shown in Tab. 4, MonoPCA achieves new SoTA performance on the unified $Omni3D_{OUT}$ split, reaching 35.0% AP and outperforming the recent foundation-based DetAny3D by $+2.8\%$. It also surpasses the best prior method on both the

Table 4: Omni3D results. To the best of our knowledge, MonoPCA is **the first model** that achieves both SoTA performance under $\text{Omni3D}_{\text{OUT}}$ benchmark and KITTI leaderboard, showcasing its superior 3D detection performance. [Key: **Best**, Second Best, **Gain**]

Method	$\text{Omni3D}_{\text{OUT}}$						KITTI Leaderboard		
	$AP_{3D}^{kit}\uparrow$	$AP_{3D}^{nus}\uparrow$	$AP_{3D}^{out}\uparrow$	$AP_{3D}^{large}\uparrow$	$AP_{3D}^{small}\uparrow$	$AP_{3D}^{car}\uparrow$	Easy↑	Mod.↑	Hard↑
MonoDGP [62]	–	–	–	–	–	–	26.35	18.72	15.97
MonoCoP [94]	–	–	–	–	–	–	27.54	19.11	16.33
ImVoxelNet [68]	23.5	23.4	21.5	–	–	–	17.15	10.97	9.15
SMOKE [45]	25.9	20.4	20.0	–	–	–	14.03	9.76	7.84
Cube R-CNN [3]	36.0	32.7	31.9	37.6	21.8	72.9	23.59	15.01	12.56
DetAny3D [88]	35.8	33.9	32.2	–	–	–	26.89	18.67	15.48
MonoPCA	**38.1**	**37.0**	**35.0**	**38.8**	**25.4**	**77.0**	**30.34**	**20.55**	**17.84**
Δ	**+2.1**	**+3.1**	**+2.8**	**+1.2**	**+3.6**	**+4.1**	**+2.80**	**+1.44**	**+1.51**

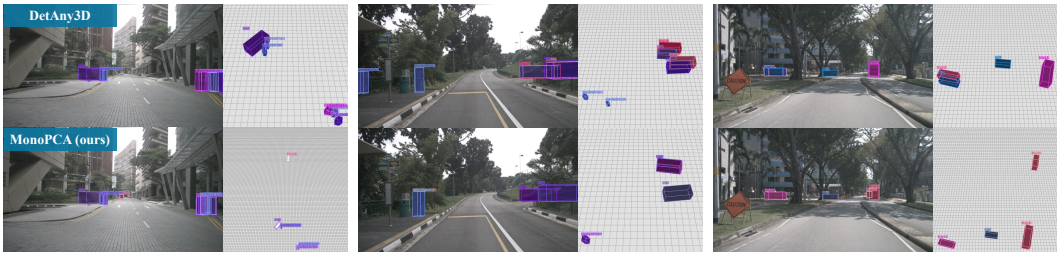


Figure 6: Qualitative comparison on $\text{Omni3D}_{\text{OUT}}$. 3D predictions overlaid on the image (Left) and the corresponding bird's-eye view with a 1m×1m grid (Right).

$\text{Omni3D}_{\text{KITTI}}$ and $\text{Omni3D}_{\text{nuScenes}}$ subsets (+2.1% and +3.1%), indicating that the improvements do not stem from dataset-specific tuning. At the category level, MonoPCA obtains +4.1% on *Car*, the dominant class in outdoor scenes, and +3.6% on small objects. Per-category results are provided in Appendix E. Together with the KITTI leaderboard results above, MonoPCA is, to the best of our knowledge, the first monocular framework to reach SoTA on both benchmarks.

Qualitative Analysis. We further provide qualitative detection results in Fig. 6. Compared with DetAny3D [88], MonoPCA produces more accurate detections for distant objects as well as small objects such as pedestrians. Additional visualization results are in Appendix F.

4.3 Ablation Study

We conduct ablation studies on the KITTI Val split, reporting AP_{3D} under $IoU_{3D} \geq 0.7$ with the Mod. setting as the primary metric. Additional results are provided in Appendix G.

Component Analysis. Tab. 5a reports the contribution of each proposed component. PCDepth and GeoAlign alone improve the baseline by +1.09% and +0.75%, respectively. Combined, they yield a +2.24% gain, exceeding the sum of the individual gains (+1.84%). The two components are therefore complementary by design: PCDepth couples depth to the predicted size and orientation, making attribute accuracy the new bottleneck, and GeoAlign addresses this bottleneck by distilling VFM-enriched representations into the student.

PCDepth Design. Tab. 5b compares different depth formulations. Direct regression achieves 23.82%, indicating that depth prediction without geometric grounding is suboptimal. GeoDepth with the measurable h_{2D} improves the result to 24.73%, validating projection-based modeling, but inherits the front-surface bias since h_{2D} encodes the projected front face rather than the object center. Replacing h_{2D} with the bias-free h'_{2D} drops performance to 24.18%, because h'_{2D} is depth-dependent and not directly observable, making its regression unstable. PCDepth retains the measurable h_{2D} and applies an orientation- and size-aware correction, preserving observability while restoring geometric consistency, and achieves the best result of 26.22% (+1.49% over GeoDepth).

MoB Design. Tab. 5c evaluates the design choice of MoB. Removing SM drops performance to 24.97% (−1.25%), while removing SA drops it to 25.44% (−0.78%). SM has a larger impact because the VFM produces only a single-scale feature; without an FPN-style expansion, it cannot be aligned

Table 5: Ablations on KITTI val (AP_{3D} Mod., $IoU_{3D} \geq 0.7$). **(a)** PCDepth and GeoAlign are complementary: their joint gain ($+2.24\%$) exceeds the sum of individual gains ($+1.09\% + 0.75\% = 1.84\%$). **(b)** Substituting the unobservable h'_{2D} degrades performance; PCDepth retains the observable h_{2D} and restores projection consistency through an analytic correction. **(c)** MoB outperforms the VFM-only variant despite a smaller backbone, indicating that convolutional features and VFM features carry complementary information. **(d)** Cosine alignment outperforms element-wise regression losses for distillation between heterogeneous backbones.

(a) Components			**(b) Depth Formulation**		**(c) MoB**		**(d) Alignment Loss**	
PCDepth	GeoAlign	Mod.↑	Formulation	Mod.↑	Configuration	Mod.↑	Loss	Mod.↑
–	–	23.98	Direct Reg.	23.82	MoB (SA only)	24.97	None	25.07
√	–	25.07	GeoDepth (h_{2D})	24.73	MoB (SM only)	25.44	MAE	25.53
–	√	24.73	GeoDepth (h'_{2D})	24.18	VFM only	25.78	MSE	25.69
√	√	**26.22**	**PCDepth**	**26.22**	**MoB (full)**	**26.22**	**HFA**	**26.22**

with the convolutional pyramid at strides $\{1/8, 1/16, 1/32, 1/64\}$ for scale-wise fusion. MoB (full) at 26.22% also outperforms a VFM-only encoder (25.78%) by $+0.44\%$ and the ResNet-50 backbone (25.07%) by $+1.15\%$, indicating that convolutional features and VFM features carry complementary information rather than redundant capacity.

Alignment Loss Design. Tab. 5d compares alignment objectives. Both MAE (25.53%) and MSE (25.69%) improve over the no-distillation baseline (25.07%), but HFA achieves the best result of 26.22%. We attribute the gap to feature-magnitude mismatch: the teacher includes VFM-derived components whose magnitudes a ResNet student cannot reproduce, so element-wise regression losses pull the student toward an unattainable scale. HFA uses cosine similarity, which is invariant to magnitude and aligns feature directions patch-wise across all four scales, making it suitable for distillation between heterogeneous backbones.

Generality across detectors. We plug PCDepth into three structurally distinct Mono3D detectors, MonoDETR [89], MonoDGP [62], and MonoCoP [94], by replacing only the depth head and jointly training the full model under each baseline's original recipe. As shown in Fig. 7, PCDepth improves the three baselines by $+1.54\%$, $+1.37\%$, and $+1.09\%$, pushing each above its published result. The gains are positive across detectors with distinct depth-modeling designs, indicating that the improvement comes from correcting the projection bias itself rather than from architecture-specific interactions. The gain also shrinks monotonically as the baseline gets stronger ($20.61 \rightarrow 22.34 \rightarrow 23.98$), reflecting the diminishing headroom toward the performance ceiling. Even on top of the strongest baseline, however, PCDepth still extracts a non-trivial $+1.09\%$, showing that the front-surface bias remains an unresolved bottleneck.

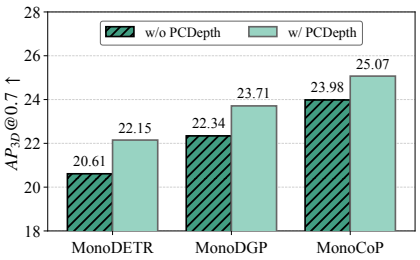


Figure 7: PCDepth generalizes across detectors. Replacing only the depth head with PCDepth yields consistent gains, indicating that the improvement stems from correcting the projection bias rather than architectural interactions.

5 Conclusion

We identify a systematic, projection-induced bias in the conventional geometric depth formulation: substituting the 2D bounding-box height h_{2D} recovers the depth of the visible front surface rather than the object center. We address this with MonoPCA, which corrects the bias analytically via PCDepth and supplies the attribute fidelity it requires through GeoAlign, a teacher-student distillation strategy that transfers VFM-enriched geometric representations into a lightweight student. MonoPCA achieves SoTA performance on KITTI, nuScenes, Waymo, and Omni3D without additional inference cost. A consequence of this formulation is that depth is no longer regressed in isolation but coupled in closed form to predicted size and orientation, making attribute accuracy a direct factor in depth quality. We hope this work motivates further study of projection-consistent depth modeling in Mono3D.

References

[1] Wentao Bao, Qi Yu, and Yu Kong. Object-aware centroid voting for monocular 3D object detection. In *IROS*, 2020. 3

[2] Ivan Barabanau, Alexey Artemov, Evgeny Burnaev, and Vyacheslav Murashkin. Monocular 3D object detection via geometric reasoning on keypoints. In *VISAPP*, 2020. 3

[3] 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. 3, 7, 9, 22, 24, 25, 26, 27, 32, 34

[4] Garrick Brazil and Xiaoming Liu. M3D-RPN: Monocular 3D region proposal network for object detection. In *ICCV*, 2019. 2, 3

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

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 7, 8, 34

[7] Yingjie Cai, Buyu Li, Zeyu Jiao, Hongsheng Li, Xingyu Zeng, and Xiaogang Wang. Monocular 3D object detection with decoupled structured polygon estimation and height-guided depth estimation. In *AAAI*, 2020. 3

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

[9] 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

[10] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. MonoRUn: Monocular 3D object detection by reconstruction and uncertainty propagation. In *CVPR*, 2021. 3

[11] Qiang Chen, Xiaokang Chen, Jian Wang, Shan Zhang, Kun Yao, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang. Group DETR: Fast DETR training with group-wise one-to-many assignment. In *ICCV*, 2023. 3

[12] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3D object detection for autonomous driving. In *CVPR*, 2016. 7

[13] Yiwei Chen, Yuguang Yao, Yihua Zhang, Bingquan Shen, Gaowen Liu, and Sijia Liu. Safety mirage: How spurious correlations undermine VLM safety fine-tuning and can be mitigated by machine unlearning. In *ICLR*, 2026. 28

[14] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023. 25

[15] Zhiyu Chong, Xinzhu Ma, Hong Zhang, Yuxin Yue, Haojie Li, Zhihui Wang, and Wanli Ouyang. MonoDistill: Learning spatial features for monocular 3D object detection. In *ICLR*, 2022. 3

[16] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3D object detection. In *CVPR*, 2020. 3

[17] Junsheng Du, Zhaocheng He, and Yuhuan Lu. RECO: Region-aware compensation for extrinsic perturbations in roadside 3D detection. In *ECCV*, 2026. 3

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 7, 8, 34

[19] Shengxi Gui, Shuang Song, Rongjun Qin, and Yang Tang. Remote sensing object detection in the deep learning era—a review. *Remote Sensing*, 2024. 3

[20] Zixuan Hu, Dongxiao Li, Xinzhu Ma, Shixiang Tang, Xiaotong Li, Wenhan Yang, and Ling-Yu Duan. Adaptive dual uncertainty optimization: Boosting monocular 3D object detection under test-time shifts. In *ICCV*, 2025. 3

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

[22] Rui Huang, Henry Zheng, Yan Wang, Zhuofan Xia, Marco Pavone, and Gao Huang. Training an open-vocabulary monocular 3D object detection model without 3D data. In *NeurIPS*, 2024. 3

[23] Changwoo Jeon, Rishi Upadhyay, and Achuta Kadambi. MoCA3D: Monocular 3D bounding box prediction in the image plane. In *ECCV*, 2026. 3

[24] 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. 7, 19, 20, 21

[25] Xueying Jiang, Sheng Jin, Lewei Lu, Xiaoqin Zhang, and Shijian Lu. Weakly supervised monocular 3D detection with a single-view image. In *CVPR*, 2024. 3

[26] Xueying Jiang, Sheng Jin, Xiaoqin Zhang, Ling Shao, and Shijian Lu. MonoMAE: Enhancing monocular 3D detection through depth-aware masked autoencoders. In *NeurIPS*, 2024. 7, 19, 20

[27] Junyoung Jung, Seokwon Kim, and Jung Uk Kim. MonoSAOD: Monocular 3D object detection with sparsely annotated label. In *CVPR*, 2026. 3

[28] Robin Kim, Colin Samplawski, and Benjamin M. Marlin. Probing the 3D object-level understanding of pre-trained detection transformers. In *ECCV*, 2026. 3

[29] Youngseok Kim, Sanmin Kim, Sangmin Sim, Jun Won Choi, and Dongsuk Kum. Boosting monocular 3D object detection with object-centric auxiliary depth supervision. *T-ITS*, 2023. 3

[30] Akshay Krishnan, Abhijit Kundu, Kevis-Kokitsi Maninis, James Hays, and Matthew Brown. OmniNOCS: A unified NOCS dataset and model for 3D lifting of 2D objects. In *ECCV*, 2024. 3

[31] 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, 7, 8, 19, 20, 21

[32] Abhinav Kumar, Garrick Brazil, and Xiaoming Liu. GrooMeD-NMS: Grouped mathematically differentiable NMS for monocular 3D object detection. In *CVPR*, 2021. 2, 3

[33] Abhinav Kumar, Yuliang Guo, Xinyu Huang, Liu Ren, and Xiaoming Liu. SeaBird: Segmentation in bird's view with dice loss improves monocular 3D detection of large objects. In *CVPR*, 2024. 3

[34] Abhinav Kumar, Yuliang Guo, Zhihao Zhang, Xinyu Huang, Liu Ren, and Xiaoming Liu. CHARM3R: Towards unseen camera height robust monocular 3D detector. In *ICCV*, 2025. 3

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

[36] Seokyeong Lee, Sithu Aung, Junyong Choi, Seungryong Kim, Ig-Jae Kim, and Junghyun Cho. PLOT: Pseudo-labeling via object tracking for monocular 3D object detection. In *ECCV*, 2026. 3

[37] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022. 6

[38] Zhuoling Li, Zhan Qu, Yang Zhou, Jianzhuang Liu, Haoqian Wang, and Lihui Jiang. Diversity matters: Fully exploiting depth clues for reliable monocular 3D object detection. In *CVPR*, 2022. 3

[39] Hongbin Lin, Zilu Guo, Yifan Zhang, Shuaicheng Niu, Yafeng Li, Ruimao Zhang, Shuguang Cui, and Zhen Li. DriveGEN: Generalized and robust 3D detection in driving via controllable text-to-image diffusion generation. In *CVPR*, 2025. 3

[40] Hongbin Lin, Yifan Zhang, Shuaicheng Niu, Shuguang Cui, and Zhen Li. MonoTTA: Fully test-time adaptation for monocular 3D object detection. In *ECCV*, 2024. 3

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

[42] Hou-I Liu, Christine Wu, Jen-Hao Cheng, Wenhao Chai, Shian-Yun Wang, Gaowen Liu, Hugo Latapie, Jhih-Ciang Wu, Jenq-Neng Hwang, Hong-Han Shuai, et al. MonoTAKD: Teaching assistant knowledge distillation for monocular 3D object detection. In *CVPR*, 2025. 3, 7, 19, 20

[43] Xianpeng Liu, Nan Xue, and Tianfu Wu. Learning auxiliary monocular contexts helps monocular 3D object detection. In *AAAI*, 2022. 7, 19, 20

[44] Xianpeng Liu, Ce Zheng, Kelvin B 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

[45] Zechen Liu, Zizhang Wu, and Roland Tóth. SMOKE: Single-stage monocular 3D object detection via keypoint estimation. In *CVPR Workshop*, 2020. 3, 9

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

[47] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3D object detection. In *ICCV*, 2021. 1, 3, 4, 8, 19, 20, 21, 24

[48] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-LiDAR representation. In *ECCV*, 2020. 3

[49] Xinzhu Ma, Wanli Ouyang, Andrea Simonelli, and Elisa Ricci. 3D object detection from images for autonomous driving: a survey. *TPAMI*, 2024. 1, 3

[50] 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

[51] 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. 3

[52] 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. *TPAMI*, 2024. 1

[53] Ruiyu Mao, Baoming Zhang, Nicholas Ruozzi, and Yunhui Guo. Learnability-driven submodular optimization for active roadside 3D detection. In *CVPR*, 2026. 3

[54] 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

[55] Youngmin Oh, Hyung-Il Kim, Seong Tae Kim, and Jung Uk Kim. MonoWAD: Weather-adaptive diffusion model for robust monocular 3D object detection. In *ECCV*, 2024. 3

[56] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 3, 25

[57] Rishubh Parihar, Srinjay Sarkar, Sarthak Vora, Jogendra Nath Kundu, and R. Venkatesh Babu. Mono-Place3D: Learning 3D-aware object placement for 3D monocular detection. In *CVPR*, 2025. 3

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

[59] Hyeonjeong Park, Peixi Xiong, Xiaoqian Ruan, Dian Jia, Pei Yu, and Wei Tang. RARE: Learn to rank and retrieve for monocular 3D object detection. In *CVPR*, 2026. 3

[60] Liang Peng, Xiaopei Wu, Zheng Yang, Haifeng Liu, and Deng Cai. DID-M3D: Decoupling instance depth for monocular 3D object detection. In *ECCV*, 2022. 3

[61] Liang Peng, Junkai Xu, Haoran Cheng, Zheng Yang, Xiaopei Wu, Wei Qian, Wenxiao Wang, Boxi Wu, and Deng Cai. Learning occupancy for monocular 3D object detection. In *CVPR*, 2024. 3, 7, 19, 20

[62] Fanqi Pu, Yifan Wang, Jiru Deng, and Wenming Yang. MonoDGP: Monocular 3D object detection with decoupled-query and geometry-error priors. In *CVPR*, 2025. 1, 3, 4, 7, 8, 9, 10, 19, 20, 21, 24, 32

[63] Zengyi Qin, Jinglu Wang, and Yan Lu. MonoGRNet: A general framework for monocular 3D object detection. *TPAMI*, 2021. 3

[64] Zequn Qin and Xi Li. MonoGround: Detecting monocular 3D objects from the ground. In *CVPR*, 2022. 3

[65] Yasiru Ranasinghe, Deepti Hegde, and Vishal M Patel. MonoDiff: Monocular 3D object detection and pose estimation with diffusion models. In *CVPR*, 2024. 3

[66] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3D object detection. In *CVPR*, 2021. 3, 7

[67] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *TPAMI*, 2017. 2

[68] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. ImVoxelNet: Image to voxels projection for monocular and multi-view general-purpose 3D object detection. In *WACV*, 2022. 9

[69] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim. Geometry-based distance decomposition for monocular 3D object detection. In *ICCV*, 2021. 1, 2, 3, 19, 20, 24

[70] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. 3

[71] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel Lopez Antequera, and Peter Kontschieder. Disentangling monocular 3D object detection: From single to multi-class recognition. *TPAMI*, 2020. 1

[72] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3D object detection. In *ICCV*, 2019. 7

[73] Jan Skvrna and Lukas Neumann. MonoSOWA: Scalable monocular 3D object detector without human annotations. In *ICCV*, 2025. 3

[74] Yongzhi Su, Yan Di, Guangyao Zhai, Fabian Manhardt, Jason Rambach, Benjamin Busam, Didier Stricker, and Federico Tombari. OPA-3D: Occlusion-aware pixel-wise aggregation for monocular 3D object detection. *RA-L*, 2023. 7, 19, 20

[75] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 7, 8, 34

[76] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: Improved training techniques for CLIP at scale. *arXiv preprint arXiv:2303.15389*, 2023. 3, 7, 26, 34

[77] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 25

[78] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *CoRL*, 2021. 3

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

[80] Yifan Wang, Yian Zhao, Fanqi Pu, Xiaochen Yang, Yang Tang, Xi Chen, and Wenming Yang. SPAN: Spatial-projection alignment for monocular 3D object detection. In *CVPR*, 2026. 3

[81] Zizhang Wu, Yuanzhu Gan, Lei Wang, Guilian Chen, and Jian Pu. MonoPGC: Monocular 3D object detection with pixel geometry contexts. In *ICRA*, 2023. 3

[82] Zizhang Wu, Yuanzhu Gan, Yunzhe Wu, Ruihao Wang, Xiaoquan Wang, and Jian Pu. FD3D: Exploiting foreground depth map for feature-supervised monocular 3D object detection. In *AAAI*, 2024. 3, 7, 19, 20

[83] Longfei Yan, Pei Yan, Shengzhou Xiong, Xuanyu Xiang, and Yihua Tan. MonoCD: Monocular 3D object detection with complementary depths. In *CVPR*, 2024. 1, 7, 19, 20

[84] Sunghun Yang, Minhyeok Lee, Jungho Lee, and Sangyoun Lee. MonoCLUE: Object-aware clustering enhances monocular 3D object detection. In *AAAI*, 2026. 3

[85] Yung-Hsu Yang, Luigi Piccinelli, Mattia Segu, Siyuan Li, Rui Huang, Yuqian Fu, Marc Pollefeys, Hermann Blum, and Zuria Bauer. 3D-MOOD: Lifting 2D to 3D for monocular open-set object detection. In *ICCV*, 2025. 3

[86] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 3

[87] Guowen Zhang, Junsong Fan, Liyi Chen, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. General geometry-aware weakly supervised 3D object detection. In *ECCV*, 2024. 3

[88] Hanxue Zhang, Haoran Jiang, Qingsong Yao, Yanan Sun, Renrui Zhang, Hao Zhao, Hongyang Li, Hongzi Zhu, and Zetong Yang. Detect anything 3D in the wild. In *ICCV*, 2025. 9

[89] Renrui Zhang, Han Qiu, Tai Wang, Ziyu Guo, Ziteng Cui, Yu Qiao, Hongsheng Li, and Peng Gao. MonoDETR: Depth-guided transformer for monocular 3D object detection. In *ICCV*, 2023. 1, 3, 4, 7, 8, 10, 16, 18, 19, 20, 21, 32

[90] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3D object detection. In *CVPR*, 2021. 1, 2, 3, 19, 20

[91] Zhihao Zhang, Shengcao Cao, and Yu-Xiong Wang. TAMM: TriAdapter multi-modal learning for 3D shape understanding. In *CVPR*, 2024. 3

[92] Zhihao Zhang, Yiwei Chen, Weizhan Zhang, Caixia Yan, Qinghua Zheng, Qi Wang, and Wangdu Chen. Tile classification based viewport prediction with multi-modal fusion transformer. In *ACM MM*, 2023. 22

[93] Zhihao Zhang, Haipeng Du, Shouqin Huang, Weizhan Zhang, and Qinghua Zheng. VRFormer: 360-degree video streaming with FoV combined prediction and super resolution. In *ISPA*, 2022. 22

[94] Zhihao Zhang, Abhinav Kumar, Girish Chandar Ganesan, and Xiaoming Liu. Unleashing the power of chain-of-prediction for monocular 3D object detection. In *CVPR*, 2026. 1, 3, 7, 8, 9, 10, 16, 19, 20, 21, 30, 32, 34

[95] Zhihao Zhang, Abhinav Kumar, and Xiaoming Liu. Towards intrinsic-aware monocular 3D object detection. In *CVPR*, 2026. 3

[96] Zhihao Zhang, Gengwei Zhang, Tianlong Chen, and Xiaoming Liu. RefineAny3D: Depth refinement as semantic alignment for monocular 3D detection. In *NeurIPS*, 2026. 3

[97] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs beat YOLOs on real-time object detection. In *CVPR*, 2024. 3

[98] Yunsong Zhou, Hongzi Zhu, Quan Liu, Shan Chang, and Minyi Guo. MonoATT: Online monocular 3D object detection with adaptive token transformer. In *CVPR*, 2023. 3

[99] Jie Zhu, Girish Chandar Ganesan, and Xiaoming Liu. DepthAgent: Towards better universal depth estimation via sample-wise expert selection. *arXiv preprint arXiv:2605.23281*, 2026. 16

[100] Jie Zhu, Xiao Guo, Yiyang Su, Anil Jain, and Xiaoming Liu. FusionAgent: A multimodal agent with dynamic model selection for human recognition. In *CVPR*, 2026. 16

[101] Jie Zhu, Yiyang Su, Minchul Kim, Anil Jain, and Xiaoming Liu. A quality-guided mixture of score-fusion experts framework for human recognition. In *ICCV*, 2025. 16

[102] Jie Zhu, Yiyang Su, and Xiaoming Liu. Can textual reasoning improve the performance of MLLMs on fine-grained visual classification? In *CVPR Findings*, 2026. 27

Appendix

A Architecture and Implementation Details of MonoPCA

A.1 Model Details

We provide a detailed description of the architecture of MonoPCA. Built upon MonoCoP [94], MonoPCA adopts a more compact design, as illustrated in Fig. A1. Compared with previous methods, MonoPCA simplifies the overall framework by jointly encoding and fusing depth and visual features. As a result, the model only requires a standard visual encoder and decoder, leading to a cleaner and more efficient architecture.

Specifically, it consists of an image backbone for extracting visual features, a depth predictor for generating the depth feature map, a visual encoder that processes the fused image and depth features, and a visual decoder that further refines the encoded representations. On top of these components, several prediction heads are employed to estimate the 3D object attributes, while the proposed PCDepth formulation is used to produce the final depth prediction.

Image Backbone. Our framework employs two types of image backbones. The teacher model adopts the proposed MoB, whereas the student model uses a standard ResNet backbone. Although their internal architectures differ, both backbones share the same input–output interface. Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, the backbone extracts multi-scale visual representations:

$$\{F_1, F_2, F_3, F_4\} = \mathcal{B}(I), \tag{A1}$$

where $\mathcal{B}$ denotes the backbone network and F_1, F_2, F_3, F_4 are feature maps with spatial resolutions of $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, and $\frac{1}{64}$ of the input image, respectively, with F_4 obtained by a stride-2 convolution on F_3. The detector consumes only F_1, F_2, F_3; F_4 is used solely by the feature alignment loss during training. MoB itself follows the broader idea of adaptively combining heterogeneous experts, which has also proven effective for human recognition [101, 100] and monocular depth estimation [99].

Depth Predictor. To obtain depth-aware representations, following [89], we employ a lightweight depth predictor that estimates depth features from the backbone outputs. Given the multi-scale feature maps $\{F_1, F_2, F_3\}$ produced by the backbone, we first resize them to a unified spatial resolution of $\frac{1}{16}$ of the input image via bilinear interpolation. The resized features are then fused through element-wise addition to aggregate multi-scale information. Formally, the fused feature is computed as

$$F_D = \phi\left(\sum_{i=1}^{3} \mathcal{R}(F_i)\right), \tag{A2}$$

where $\mathcal{R}(\cdot)$ denotes the resizing operation to the $\frac{1}{16}$ scale and $\phi(\cdot)$ represents two stacked 3×3 convolution layers. The resulting depth feature map is

$$F_D \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times C}.$$

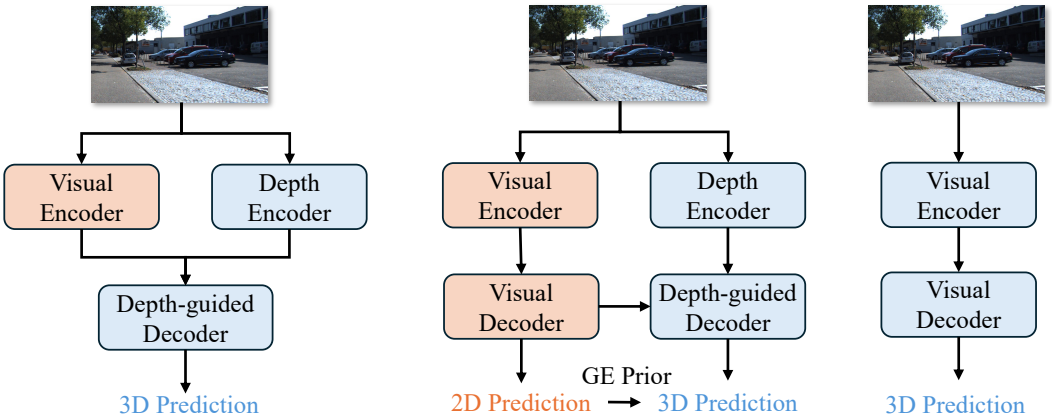
Visual Encoder. The visual encoder aims to jointly encode and fuse the visual features from the image backbone and the depth predictor. Specifically, it takes as input the three multi-scale backbone features $\{F_1, F_2, F_3\}$ together with the depth feature F_D, forming a four-level feature set. These features are then processed by a stack of three deformable attention layers to capture cross-scale interactions and integrate depth-aware cues into the visual representations. Formally, the encoded features are obtained as

$$\{E_1, E_2, E_3, E_4\} = \mathcal{E}\big(\{F_1, F_2, F_3, F_D\}\big), \tag{A3}$$

where $\mathcal{E}(\cdot)$ denotes the visual encoder composed of three deformable attention layers.

Visual Decoder. The visual decoder refines the encoded visual representations and prepares them for the final 3D attribute prediction. It takes the encoded multi-level features $\{E_1, E_2, E_3, E_4\}$ from the visual encoder as input and processes them through a transformer decoder consisting of one self-attention layer followed by one cross-attention layer. The self-attention layer models interactions among object queries, while the cross-attention layer aggregates information from the encoded visual features. Formally, the decoded representations are obtained as

$$Z = \mathcal{D}(Q, \{E_1, E_2, E_3, E_4\}), \tag{A4}$$



(a) MonoDETR / MonoCoP (b) MonoDGP (c) Ours

Figure A1: Architectural comparison between MonoPCA and previous methods. We highlight the key architectural differences between MonoPCA and representative monocular 3D detectors, including MonoCoP, MonoDGP, and MonoDETR.

where Q denotes the learnable object queries and $\mathcal{D}(\cdot)$ represents the visual decoder.

Prediction Heads. The decoded representations Z are fed into several prediction heads to estimate the 3D object attributes. Specifically, each object query predicts a set of attributes including the 2D bounding box, object size, orientation, and corresponding confidence. Each attribute is predicted by a lightweight multi-layer perceptron (MLP) head. Formally, the predicted attributes are obtained as

$$\hat{y} = \mathcal{H}(Z), \tag{A5}$$

where $\mathcal{H}(\cdot)$ denotes the collection of prediction heads.

Final Depth Estimation with PCDepth. Based on the predicted geometric attributes, the final object depth is computed using the proposed PCDepth formulation. Unlike conventional geometric depth estimation that directly relies on the projected 2D height, PCDepth incorporates object size and orientation information to obtain a projection-consistent depth estimation. Formally, the final depth is computed as

$$\hat{z} = \text{PCDepth}(\hat{s}, \hat{o}, \hat{h}), \tag{A6}$$

where $\hat{s}$, $\hat{o}$, and $\hat{h}$ denote the predicted object size, orientation, and 2D height, respectively.

A.2 Training Objective

Following DETR-style detectors, we adopt a bipartite matching strategy to assign predictions to ground-truth objects and supervise the matched pairs with multiple task-specific losses. Given the predicted object set and ground-truth annotations, we first compute an optimal matching using the Hungarian algorithm, and then apply supervision on each matched pair.

Classification Loss. For object classification, we employ the sigmoid focal loss to address the foreground–background imbalance:

$$\mathcal{L}_{cls} = \text{FocalLoss}(p, y), \tag{A7}$$

where p denotes the predicted class logits and y represents the ground-truth class label.

Bounding Box Loss. For 2D box regression, we use an ℓ_1 loss and the GIoU loss:

$$\mathcal{L}_{bbox} = \|b - b^*\|_1, \qquad \mathcal{L}_{giou} = 1 - \text{GIoU}(b, b^*), \tag{A8}$$

where b and b^* denote the predicted and ground-truth boxes, respectively.

3D Center Loss. The projected 3D object center is supervised using an ℓ_1 loss:

$$\mathcal{L}_{center} = \|c - c^*\|_1. \tag{A9}$$

Depth Loss. For depth prediction, we adopt an uncertainty-aware regression loss:

$$\mathcal{L}_{depth} = \sqrt{2}\, e^{-s}|z - z^*| + s, \tag{A10}$$

where z and z^* denote the predicted and ground-truth depths, and s represents the predicted log-variance that models depth uncertainty.

Dimension Loss. Object size prediction is supervised using a scale-normalized ℓ_1 loss:

$$\mathcal{L}_{dim} = \left\| \frac{d - d^*}{d^*} \right\|_1, \tag{A11}$$

where d and d^* denote the predicted and ground-truth object dimensions.

Orientation Loss. For object orientation, we follow a classification–regression formulation. The orientation angle is first discretized into bins for classification, followed by residual regression within each bin:

$$\mathcal{L}_{angle} = \mathcal{L}_{cls}^{angle} + \mathcal{L}_{reg}^{angle}. \tag{A12}$$

Auxiliary Depth Map Loss. To further enhance depth understanding, we follow MonoDETR [89] and supervise the predicted depth map with its depth-aware loss $\mathcal{L}_{depthmap}$.

Feature Alignment Loss. To transfer the enriched representations from the teacher network, we employ the proposed hierarchical feature alignment loss $\mathcal{L}_{align}$, which enforces patch-wise cosine similarity between teacher and student features across all four pyramid levels $\{F_1, F_2, F_3, F_4\}$:

$$\mathcal{L}_{align} = \sum_{s=1}^{4} \frac{1}{N_s} \sum_{i=1}^{N_s} \left(1 - \frac{\langle \hat{F}_{s,i}^S, F_{s,i}^T \rangle}{\|\hat{F}_{s,i}^S\|_2 \, \|F_{s,i}^T\|_2} \right), \tag{A13}$$

where F_s^T and F_s^S denote the teacher and student features at scale s, and $\hat{F}_s^S$ is the projected student feature. This loss encourages the student network to mimic the multi-scale representations produced by the teacher. The alignment loss is applied only during training, and the teacher network is discarded during inference.

Overall Loss. The final training objective is defined as

$$\begin{aligned} \mathcal{L} = {} & \lambda_{cls}\mathcal{L}_{cls} + \lambda_{bbox}\mathcal{L}_{bbox} + \lambda_{giou}\mathcal{L}_{giou} + \lambda_{center}\mathcal{L}_{center} \\ & + \lambda_{depth}\mathcal{L}_{depth} + \lambda_{dim}\mathcal{L}_{dim} + \lambda_{angle}\mathcal{L}_{angle} \\ & + \lambda_{depthmap}\mathcal{L}_{depthmap} + \lambda_{align}\mathcal{L}_{align}. \end{aligned} \tag{A14}$$

The loss weights are set as $\lambda_{cls} = 2$, $\lambda_{bbox} = 5$, $\lambda_{giou} = 2$, $\lambda_{center} = 10$, $\lambda_{dim} = 1$, $\lambda_{angle} = 1$, $\lambda_{depth} = 1$, $\lambda_{depthmap} = 1$, and $\lambda_{align} = 5$.

A.3 Implementation Details

We train the network using the AdamW optimizer with an initial learning rate of 2×10^{-4} and a weight decay of 1×10^{-4}. The teacher backbone is optimized with a smaller learning rate of 2×10^{-5}, while the student network uses a learning rate of 2×10^{-4}. The learning rate follows a step decay schedule with a decay factor of 0.5, applied at epochs 85, 125, 165, and 205, with a total of TODO training epochs. All experiments are conducted on a single node with eight NVIDIA A100 GPUs (80GB) and a batch size of 16. A full training run on KITTI takes approximately 12 hours, while training on the larger Waymo and Omni3D benchmarks takes one to two days.

B Detailed KITTI Results

Leaderboard Results. We provide detailed results on the KITTI test benchmark. In KITTI, the car category is the dominant class and the primary evaluation target. Therefore, we report the detailed leaderboard results for car detection, which is also the most challenging setting in Mono3D. As shown in Tab. A1, MonoPCA outperforms previous methods by a clear margin. For example, in the Easy setting, it improves over MonoCoP by 2.80%. MonoPCA also achieves the best performance on both the Moderate and Hard levels, demonstrating consistent improvements under increasing

Table A1: Detailed KITTI Leaderboard (Test) results at $\text{IoU}_{3D} \geq 0.7$. [Key: **Best**, T: MoB teacher]

Method	Extra Data	Venue	**Test**, AP_{3D} (↑) Easy	Mod.	Hard	**Test**, AP_{BEV} (↑) Easy	Mod.	Hard
OccupancyM3D [61]	LiDAR	CVPR 24	25.55	17.02	14.79	35.38	24.18	21.37
OPA-3D [74]	Depth	ICRA 23	24.68	17.17	14.14	32.50	23.14	20.30
MonoTAKD [42]*	LiDAR	CVPR 25	27.91	19.43	16.51	38.75	27.76	24.14
MonoFlex [90]	None	CVPR 21	19.94	13.89	12.07	28.23	19.75	16.89
MonoRCNN [69]		ICCV 21	18.36	12.65	10.03	25.48	18.11	14.10
GUP Net [47]		CVPR 21	20.11	14.20	11.77	–	–	–
DEVIANT [31]		ECCV 22	21.88	14.46	11.89	29.65	20.44	17.43
MonoCon [43]		AAAI 22	22.50	16.46	13.95	31.12	22.10	19.00
MonoDETR [89]		ICCV 23	25.00	16.47	13.58	33.60	22.11	18.60
MonoUNI [24]		NeurIPS 23	24.75	16.73	13.49	–	–	–
FD3D [82]		AAAI 24	25.38	17.12	14.50	34.20	23.72	20.76
MonoMAE [26]		NeurIPS 24	25.60	18.84	16.78	34.15	24.93	21.76
MonoCD [83]		CVPR 24	25.53	16.59	14.53	33.41	22.81	19.57
MonoDGP [62]		CVPR 25	26.35	18.72	15.97	35.24	25.23	22.02
MonoCoP [94]		CVPR 26	27.54	19.11	16.33	36.77	25.57	22.62
MonoPCA		–	**30.34**	**20.55**	**17.84**	**38.65**	**26.63**	**24.43**
MonoPCA-T	None	–	**31.78**	**21.97**	**19.24**	**40.15**	**27.64**	**25.54**

occlusion and object difficulty. Moreover, compared with methods that utilize additional data such as LiDAR or depth maps, MonoPCA still achieves higher performance on all AP_{3D} metrics.

KITTI Val Results. We further report detailed results on the KITTI Val set in Tab. A2. Among methods that do not rely on extra data, MonoPCA achieves the best performance across all evaluation metrics. Specifically, MonoPCA attains 35.86%, 26.22%, and 23.58% AP_{3D} on the Easy, Moderate, and Hard splits, outperforming the previous best method MonoCoP by 3.80%, 2.24%, and 2.94%, respectively. Consistent improvements are also observed for BEV detection, where MonoPCA achieves 46.29%, 33.89%, and 30.05% AP_{BEV} on the Easy, Moderate, and Hard settings. Furthermore, the teacher model MonoPCA-T achieves even stronger performance, reaching 37.39%, 28.44%, and 25.02% AP_{3D}, demonstrating the effectiveness of MoB.

C Detailed nuScenes Results

On the nuScenes Val set, MonoPCA achieves new SoTA performance under both IoU thresholds 0.7 and 0.5 across 3D and BEV detection. Under the primary setting $\text{IoU} \geq 0.7$, MonoPCA improves AP_{3D} Mod. by $+2.32\%$ over MonoCoP, with consistent gains in BEV detection. At $\text{IoU} \geq 0.5$, the advantage becomes more pronounced, demonstrating strong robustness under relaxed localization constraints. The teacher MonoPCA-T yields further gains in AP_{3D} under both IoU thresholds, and the distilled student recovers about half of the teacher's improvement over MonoCoP, demonstrating the effectiveness of GeoAlign.

D Detailed Waymo Results

Tabs. A4 and A5 present detailed evaluation results on the Waymo benchmark under two IoU thresholds (0.5 and 0.7), covering both Level 1 and Level 2 settings.

Results at IoU ≥ 0.5. Under the looser IoU threshold, MonoPCA already achieves clear improvements over previous monocular methods. For Level 1, MonoPCA obtains 12.17% AP_{3D} and 12.05% APH_{3D}, surpassing MonoCoP (11.76% / 11.65%) by about $+0.4$ points. In relative terms, the improvement is largest in the 30–50 m range, where MonoPCA reaches 7.64% AP_{3D}, substantially outperforming the previous best 6.03%. Beyond 50 m, it also achieves the strongest performance with 1.68%. With the high-capacity teacher architecture, MonoPCA-T further pushes the performance to 15.34% AP_{3D} and 15.23% APH_{3D} on Level 1, improving over MonoCoP by more than

Table A2: Detailed KITTI Val results at $\text{IoU}_{3D} \geq 0.7$.[Key: **Best**, T: MoB teacher]

Method	Extra Data	Venue	**Val**, AP_{3D} (↑) Easy	Mod.	Hard	**Val**, AP_{BEV} (↑) Easy	Mod.	Hard
OccupancyM3D [61]	LiDAR	CVPR 24	26.87	19.96	17.15	35.72	26.60	23.68
OPA-3D [74]	Depth	ICRA 23	24.97	19.40	16.59	33.80	25.51	22.13
MonoTAKD [42]*	LiDAR	CVPR 25	34.36	22.61	19.88	42.86	29.41	26.47
MonoFlex [90]	None	CVPR 21	23.64	17.51	14.83	–	–	–
MonoRCNN [69]		ICCV 21	–	–	–	–	–	–
GUP Net [47]		CVPR 21	22.76	16.46	13.72	31.07	22.94	19.75
DEVIANT [31]		ECCV 22	24.63	16.54	14.52	32.60	23.04	19.99
MonoCon [43]		AAAI 22	26.33	19.01	15.98	–	–	–
MonoDETR [89]		ICCV 23	28.84	20.61	16.38	37.86	26.95	22.80
MonoUNI [24]		NeurIPS 23	24.51	17.18	14.01	–	–	–
FD3D [82]		AAAI 24	28.22	20.23	17.04	36.98	26.77	23.16
MonoMAE [26]		NeurIPS 24	30.29	20.90	17.61	40.26	27.08	23.14
MonoCD [83]		CVPR 24	26.45	19.37	16.38	34.60	24.96	21.51
MonoDGP [62]		CVPR 25	30.76	22.34	19.02	39.40	28.20	24.42
MonoCoP [94]		CVPR 26	32.06	23.98	20.64	41.92	30.75	27.06
MonoPCA		–	**35.86**	**26.22**	**23.58**	**46.29**	**33.89**	**30.05**
MonoPCA-T	None	–	**37.39**	**28.44**	**25.02**	**47.95**	**35.36**	**31.50**

Table A3: nuScenes Val results. MonoPCA achieves SoTA performance on both 3D and BEV detection under IoU thresholds 0.7 and 0.5. [Key: **Best**, Second Best, **Gain**, T: MoB teacher]

Method	**$\text{IoU}_{3D} \geq 0.7$**				**$\text{IoU}_{3D} \geq 0.5$**			
	AP_{3D} ↑		AP_{BEV} ↑		AP_{3D} ↑		AP_{BEV} ↑	
	Easy	Mod.	Easy	Mod.	Easy	Mod.	Easy	Mod.
GUP Net [47]	8.50	7.40	14.21	12.81	29.03	26.16	33.42	30.23
DEVIANT [31]	9.69	8.33	16.28	14.36	31.47	28.22	35.61	31.93
MonoDETR [89]	9.53	8.19	16.39	14.41	31.81	28.35	35.70	31.96
MonoDGP [62]	10.04	8.78	16.55	14.53	29.56	26.17	32.67	29.44
MonoCoP [94]	10.85	9.71	17.83	15.86	33.70	29.91	37.44	34.01
MonoPCA	**13.75**	**12.03**	**20.97**	**18.49**	**38.50**	**34.11**	**42.03**	**38.49**
Δ	**+2.90**	**+2.32**	**+3.14**	**+2.63**	**+4.80**	**+4.20**	**+4.59**	**+4.48**
MonoPCA-T	**16.52**	**14.57**	**25.06**	**22.40**	**41.81**	**37.22**	**45.10**	**41.41**

+3.5 points. Notably, the teacher model achieves 8.57% AP_{3D} at 30–50 m and 1.92% beyond 50 m, establishing new state-of-the-art results across all distance ranges. A similar trend is observed for Level 2. MonoPCA reaches 11.60% AP_{3D} and 11.49% APH_{3D}, consistently outperforming previous monocular approaches. The teacher model further improves the results to 14.53% AP_{3D} and 14.42% APH_{3D}, again achieving the best performance across all distance splits.

Results at IoU ≥ 0.7. Under the stricter localization criterion, MonoPCA achieves the best results in the 0–30 m and 50 m+ ranges and the best AP_{3D} in the 30–50 m range. For Level 1, MonoPCA achieves 2.98% AP_{3D} and 2.96% APH_{3D}, exceeding MonoCoP (2.72% / 2.70%). The teacher model further improves the results to 3.18% AP_{3D} and 3.16% APH_{3D}, with the strongest performance across all distance ranges. The same trend holds for Level 2, where MonoPCA achieves 2.85% AP_{3D} and 2.84% APH_{3D}, while MonoPCA-T further improves to 2.98% AP_{3D} and 2.96% APH_{3D}.

Overall, MonoPCA improves detection accuracy at both evaluation levels, with gains in nearly all distance ranges. The gains are largest within 50 m, where the front-surface offset is a dominant error source; beyond this range, depth noise that grows with distance limits the benefit of any correction.

Table A4: Waymo Val Vehicle results with $\text{IoU}_{3D} \geq 0.5$. [Key: **Best**, **T**: MoB teacher]

Method	Difficulty	APH_{3D} (%)				AP_{3D} (%)			
		All	0-30	30-50	50-∞	All	0-30	30-50	50-∞
GUP Net [47] in [31]	Level 1	9.94	24.59	4.78	0.22	10.02	24.78	4.84	0.22
DEVIANT [31]		10.89	26.64	5.08	0.18	10.98	26.85	5.13	0.18
MonoDETR [89]		9.60	23.58	4.67	0.99	9.68	23.78	4.72	1.00
MonoUNI [24]		10.73	26.30	3.98	0.55	10.98	26.63	4.04	0.57
MonoDGP [62]		9.84	23.73	5.01	0.98	10.06	24.01	5.06	0.99
MonoCoP [94]		11.65	27.35	5.97	1.46	11.76	27.59	6.03	1.48
MonoPCA		12.05	31.99	7.59	1.66	12.17	32.36	7.64	1.68
MonoPCA-T		**15.23**	**36.52**	**8.50**	**1.90**	**15.34**	**36.79**	**8.57**	**1.92**
GUP Net [47] in [31]	Level 2	9.31	24.50	4.62	0.19	9.39	24.69	4.67	0.19
DEVIANT [31]		10.20	26.54	4.90	0.16	10.29	26.75	4.95	0.16
MonoDETR [89]		9.00	23.49	4.51	0.86	9.08	23.70	4.55	0.87
MonoUNI [24]		10.24	26.24	3.89	0.51	10.38	26.57	3.95	0.53
MonoDGP [62]		9.32	23.65	4.84	0.85	9.43	23.92	4.88	0.86
MonoCoP [94]		10.93	27.25	5.76	1.27	11.03	27.49	5.82	1.29
MonoPCA		11.49	31.91	7.43	1.42	11.60	32.27	7.48	1.44
MonoPCA-T		**14.42**	**36.39**	**8.25**	**1.67**	**14.53**	**36.66**	**8.31**	**1.69**

Table A5: Waymo Val Vehicle results with $\text{IoU}_{3D} \geq 0.7$. [Key: **Best**, **T**: MoB teacher]

Method	Difficulty	APH_{3D} (%)				AP_{3D} (%)			
		All	0-30	30-50	50-∞	All	0-30	30-50	50-∞
GUP Net [47] in [31]	Level 1	2.27	6.11	0.80	0.03	2.28	6.15	0.81	0.03
DEVIANT [31]		2.67	6.90	0.98	0.02	2.69	6.95	0.99	0.02
MonoDETR [89]		2.10	5.94	0.73	0.12	2.11	5.99	0.73	0.12
MonoUNI [24]		3.16	8.50	0.86	0.12	3.20	8.61	0.87	0.13
MonoDGP [62]		2.39	6.62	0.84	0.12	2.41	6.67	0.84	0.12
MonoCoP [94]		2.70	7.38	1.06	0.16	2.72	7.44	1.07	0.16
MonoPCA		2.96	9.15	1.04	0.20	2.98	9.21	1.12	0.20
MonoPCA-T		**3.16**	**10.30**	**1.26**	**0.22**	**3.18**	**10.36**	**1.27**	**0.22**
GUP Net [47] in [31]	Level 2	2.12	6.08	0.77	0.02	2.14	6.13	0.78	0.02
DEVIANT [31]		2.50	6.87	0.94	0.02	2.52	6.93	0.95	0.02
MonoDETR [89]		1.97	5.92	0.70	0.10	1.98	5.96	0.71	0.10
MonoUNI [24]		3.00	8.48	0.84	0.12	3.04	8.59	0.85	0.12
MonoDGP [62]		2.24	6.59	0.81	0.10	2.26	6.65	0.81	0.10
MonoCoP [94]		2.53	7.35	1.02	0.14	2.55	7.41	1.03	0.14
MonoPCA		2.84	9.13	1.00	0.17	2.85	9.18	1.11	0.17
MonoPCA-T		**2.96**	**10.26**	**1.22**	**0.19**	**2.98**	**10.33**	**1.23**	**0.19**

E Additional $\text{Omni3D}_{\text{OUT}}$ Results

While the Omni3D benchmark reports only the averaged AP_{3D} across all categories, we additionally provide per-category results for completeness. For KITTI, we report performance across its five evaluated categories (see Tab. A6), and for nuScenes, we include all nine object categories in the benchmark (see Tab. A7). Per-category results of the teacher MonoPCA-T are also reported.

We also present a radar chart (see Fig. A2) to illustrate the overall performance of MonoPCA across ten evaluation benchmarks. As shown in the figure, MonoPCA consistently outperforms previous

Table A6: AP_{3D} performance across 5 categories on KITTI. [Key: **Best**, **T**: MoB teacher]

Method	AP_{3D} on Category					
	Cyclist	Truck	Pedestrian	Van	Car	Avg
MonoPCA	28.85	27.69	36.29	21.22	76.35	38.09
MonoPCA-T	27.68	27.25	38.23	28.62	76.18	39.60

Table A7: AP_{3D} performance across 9 categories on nuScenes. [Key: **Best**, **T**: MoB teacher]

Method	AP_{3D} on Category									
	Pedestrian	Motorcycle	Truck	Trailer	Car	Traffic Cone	Bus	Barrier	Bicycle	Avg
MonoPCA	14.45	24.62	45.04	26.93	77.64	11.52	70.52	40.57	21.53	36.99
MonoPCA-T	15.15	27.62	44.41	27.39	78.93	17.18	67.98	51.04	23.01	39.20

state-of-the-art methods across diverse datasets, including KITTI, nuScenes, Waymo, and Omni3D, demonstrating its strong generalization ability under different evaluation settings.

F Additional Visualization Results

Following [3], we visualize the 3D predictions overlaid on the input image (Left) and the corresponding bird's eye view with a 1m $\times$ 1m ground grid (Right). MonoPCA produces more accurate predictions, particularly for distant and small objects. Additional qualitative examples on the KITTI dataset are provided in Fig. A3 and Fig. A4. Qualitative results on the nuScenes dataset are shown in Fig. A5 and Fig. A6.

G Additional Ablation Study

G.1 Things We Tried That Did Not Make it into the Main Algorithm

From Deformable Attention to Full Attention. Following MonoCoP, we adopt deformable attention in both the visual encoder and the decoder. Compared with vanilla Transformer attention, deformable attention is significantly more efficient, as it attends to a sparse set of sampled key locations rather than all spatial tokens. However, this design also limits the ability to capture global context. Motivated by this observation, and by the success of full attention in modeling long-range and cross-modal dependencies in other domains [92, 93], we explored replacing deformable attention with standard full attention in both the encoder and decoder to enable global feature interactions.

In practice, however, this modification did not lead to improvements. Using full attention substantially increases the computational cost and memory consumption due to its quadratic complexity with respect to the number of tokens. More importantly, we observed a consistent performance drop compared with the deformable attention design. We conjecture that the sparse sampling mechanism in deformable attention provides an implicit inductive bias that helps the model focus on geometrically relevant regions, whereas full attention introduces excessive global interactions that may dilute these spatial cues. Therefore, we retain the deformable attention architecture in the final design due to its superior efficiency and better empirical performance.

Predicting h'_{2D} for GeoDepth. To eliminate the projection-induced bias in the GeoDepth formulation, a straightforward idea is to directly predict the projective height h'_{2D} and use it in place of the conventional 2D bounding box height h_{2D}. This modification is conceptually simple and can correct the bias in the geometric relation.

However, in practice this approach does not improve detection performance and often leads to unstable training. We observe that h'_{2D} is inherently depth-dependent and varies with object pose and occlusion, making it difficult to estimate reliably from image features alone. As a result, direct regression of h'_{2D} introduces additional prediction uncertainty, which propagates to the depth estimation and degrades the final 3D detection accuracy. Therefore, instead of directly predicting

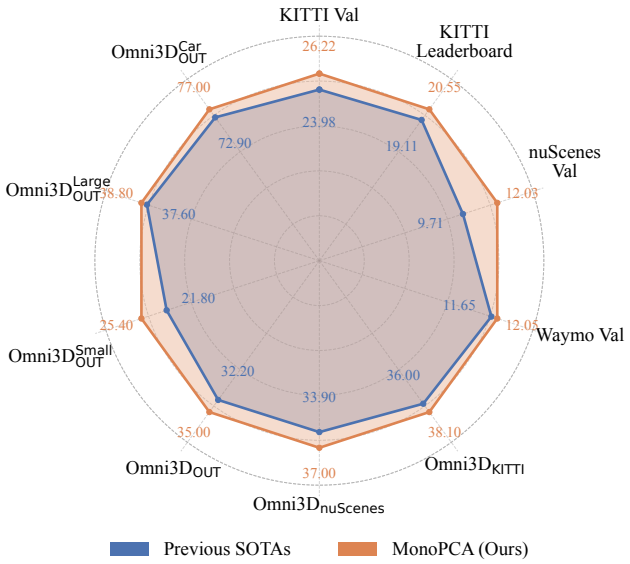


Figure A2: MonoPCA across ten benchmarks.

h'_{2D}, our proposed formulation models the geometric correction explicitly and in closed form through PCDepth, which achieves more stable optimization and better empirical performance.

Integrating VFM without FPN. Another design we explored is to directly integrate the last-layer features from the Vision Foundation Model (VFM) without constructing a multi-scale feature pyramid. This approach simplifies the architecture and reduces additional feature processing, as the detection head operates directly on the high-level semantic representation produced by the VFM.

However, we find that relying on a single-scale feature significantly degrades detection performance. Object detection, especially in Mono3D settings, benefits from multi-scale representations to capture objects at different sizes and distances. Using only the final VFM feature map removes this multi-scale structure and weakens the model's ability to localize objects accurately. Therefore, in the final design we retain the FPN-based multi-scale feature construction when integrating VFM features, which provides richer spatial representations and leads to consistently better detection performance.

Using Pixel Shuffle for Token Reduction. We also explored reducing the number of image tokens using a pixel-shuffle style operation. Specifically, neighboring feature maps are first merged and then projected through a lightweight projector to reduce the channel dimension. This operation effectively decreases the number of image tokens and thus lowers the computational cost of subsequent transformer layers.

However, we observe that this design consistently leads to a noticeable performance drop, typically around $2\% \sim 3\%$ in detection accuracy. We conjecture that merging neighboring features in this manner removes fine-grained spatial details that are important for accurate object localization in Mono3D detection. Therefore, we do not adopt this token reduction strategy.

Replacing PositionEmbeddingSine with RoPE. We also explored replacing the commonly used PositionEmbeddingSine with Rotary Position Embedding (RoPE), which has been widely adopted in Transformer architectures for its ability to encode relative positional information.

However, this modification does not improve detection performance in our setting. In practice, we observe a slight performance degradation compared with the original sine positional embedding. We conjecture that RoPE is more suitable for sequence modeling tasks, while dense prediction tasks such as Mono3D detection rely more heavily on explicit spatial priors. The PositionEmbeddingSine embedding provides a stable and well-structured spatial encoding for image features, which appears to be more effective for our detection framework. As a result, we retain PositionEmbeddingSine in the final model design.



Figure A3: Qualitative results on Omni3D$_{\text{KITTI}}$. Following [3], we show the 3D predictions overlaid on the image (Left), and the corresponding top view with a 1m$\times$1m ground grid (Right). MonoPCA presents more accurate results for **distant and small objects.**

G.2 More Depth Design

Tab. A8 further compares different depth design strategies. We evaluate several representative formulations, including keypoint-based depth prediction [69], 2D height uncertainty modeling [47], and bias regression [62]. As shown in the table, progressively improving the depth formulation consistently enhances detection performance. Keypoint prediction achieves 30.73%, 23.25%, and 19.85% on Easy, Moderate, and Hard, while introducing 2D height uncertainty improves the results to 32.52%, 25.06%, and 21.19%. Regressing the bias further increases the performance to 33.84%,

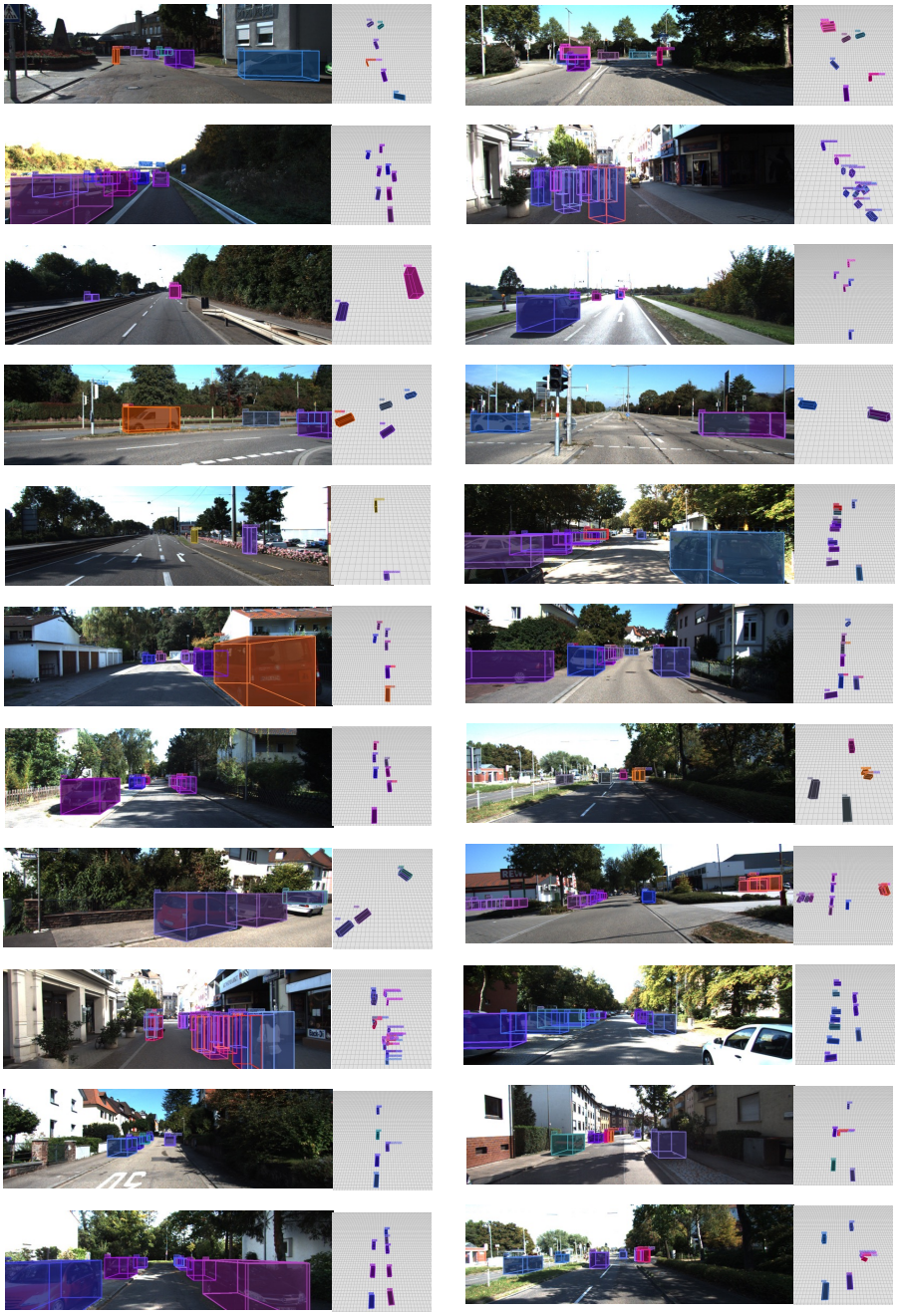


Figure A4: Qualitative results on $\text{Omni3D}_{\text{KITTI}}$. Following [3], we show the 3D predictions overlaid on the image (Left), and the corresponding top view with a 1m×1m ground grid (Right). MonoPCA presents more accurate results for **distant and small objects.**

25.73%, and 22.30%. Our **PCDepth** achieves the best results with 35.86%, 26.22%, and 23.58%, demonstrating that a more principled depth formulation leads to more accurate depth estimation and better 3D detection performance.

G.3 VFM Selection

Tab. A9 presents the ablation study on the choice of Vision Foundation Models (VFMs). We compare several representative pretrained encoders, including DINOv2 [56], SigLIP 2 [77], OpenCLIP [14],



Figure A5: Qualitative results on Omni3D$_{\text{nuScenes}}$. Following [3], we show the 3D predictions overlaid on the image (Left), and the corresponding top view with a 1m$\times$1m ground grid (Right). MonoPCA presents more accurate results for **distant and small objects.**

and EVA-CLIP [76]. Among them, DINOv2 achieves 34.73%, 26.05%, and 22.55% on Easy, Moderate, and Hard settings, respectively, while SigLIP 2 is lower on all three. OpenCLIP attains higher Easy and Hard scores than DINOv2 (35.26% and 22.71%) but a lower Moderate score (25.31%). EVA-CLIP delivers the best performance across all difficulty levels, achieving 35.86%, 26.22%, and 23.58%, demonstrating the effectiveness of stronger pretrained visual representations for our framework. Therefore, EVA-CLIP is adopted as the default VFM in the main experiments. We note that stronger vision–language pretraining does not automatically translate into fine-grained perception, as multimodal LLMs built on such encoders still struggle with fine-grained visual

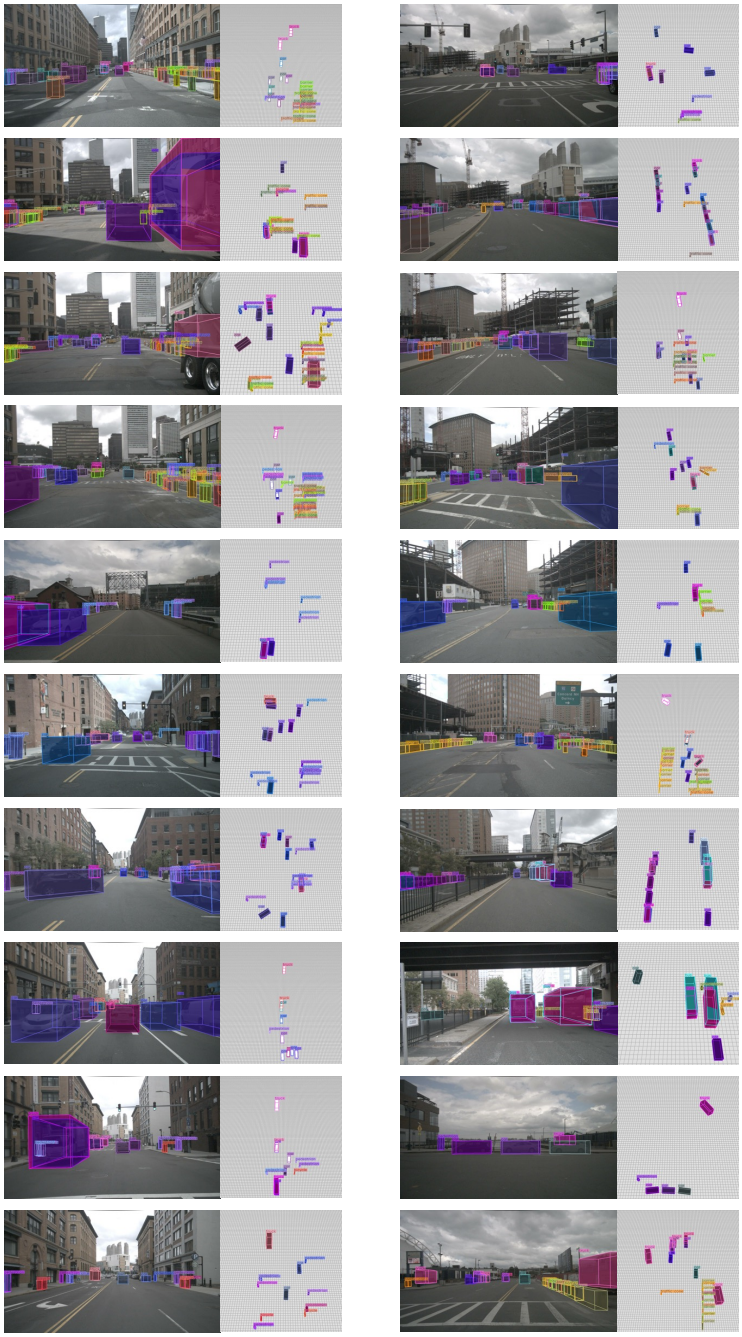


Figure A6: Qualitative results on $\text{Omni3D}_{\text{nuScenes}}$. Following [3], we show the 3D predictions overlaid on the image (Left), and the corresponding top view with a 1m×1m ground grid (Right). MonoPCA presents more accurate results for **distant and small objects.**

classification [102]; this motivates distilling VFM features into a task-specific student rather than relying on the VFM alone.

G.4 Alignment Loss Coefficient

Tab. A10 studies the effect of the HFA loss coefficient λ_{align}. We vary the coefficient from 1 to 10 and report the detection performance under different difficulty levels. As the coefficient increases from 1 to 5, the performance consistently improves, reaching the best results at 5 with 35.86%, 26.22%, and

Table A8: Ablation on depth design.

More Depth Design	Easy↑	Mod.↑	Hard↑
Keypoint Prediction	30.73	23.25	19.85
2D Height Uncertainty	32.52	25.06	21.19
Regress Bias	33.84	25.73	22.30
PCDepth (Ours)	**35.86**	**26.22**	**23.58**

Table A9: Ablation on VFM selection.

VFM Selection	Easy↑	Mod.↑	Hard↑
DINOv2	34.73	26.05	22.55
SigLIP 2	32.99	25.51	22.04
OpenCLIP	35.26	25.31	22.71
EVA-CLIP (Ours)	**35.86**	**26.22**	**23.58**

Table A10: Ablation on HFA coefficient.

HFA Coefficient	Easy↑	Mod.↑	Hard↑
1	33.46	25.06	21.90
2	34.73	25.94	22.12
5	**35.86**	**26.22**	**23.58**
10	33.89	25.21	21.91

23.58% on Easy, Moderate, and Hard, respectively. However, further increasing the coefficient to 10 leads to noticeable performance degradation. This observation suggests that a moderate coefficient provides a better balance for the distillation, and we therefore set the HFA coefficient to 5 in all experiments.

H Limitations

MonoPCA has three main limitations. First, the closed-form correction in Eq. (6) assumes pinhole projection and an upright box, and treats h_{2D} as the tight projection of the eight 3D corners; deviations from this last assumption (truncation, occlusion, annotation tightness) account for the residual error of PCDepth under oracle inputs in Fig. 1a. Second, depth accuracy is now coupled to predicted size and yaw, so residual attribute errors, especially on heavily occluded or distant objects, still propagate into depth. Third, our evaluation focuses on outdoor driving benchmarks (KITTI, nuScenes, Waymo, and Omni3D), and the behavior of MonoPCA under substantially different camera setups has not been studied.

I Broader Impact

MonoPCA targets Mono3D, which underpins applications in autonomous driving, robotics, and augmented reality. By improving depth accuracy without additional sensors or supervision, our work may extend the reach of camera-only 3D perception in settings where LiDAR is impractical or cost-prohibitive. At the same time, any 3D detection technology carries potential for misuse in unauthorized surveillance, and imperfect perception deployed in safety-critical systems can lead to harm. Moreover, because GeoAlign distills representations from a foundation model, the student may inherit spurious correlations or biases present in the teacher [13]. We position MonoPCA as research on geometric depth modeling rather than a deployment-ready system: real-world use should be paired with rigorous validation, redundant sensing where feasible, and adherence to domain-specific safety standards.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Sec. 1) precisely state our three contributions: (i) the identification of the front-surface bias in the conventional geometric depth formulation $z = fH/h_{2D}$, (ii) PCDepth, a projection-consistent depth formulation that corrects the bias in closed form, and (iii) GeoAlign, a teacher–student distillation strategy that strengthens attribute perception. Each claim is supported by experimental results in Sec. 4, including the bias diagnosis under oracle inputs (Fig. 1), SoTA results on KITTI, nuScenes, Waymo, and Omni3D, and ablation studies on each proposed component. Reported gains and the scope of evaluation faithfully match the claims made.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss three main limitations in Appendix H. First, the closed-form correction assumes pinhole projection and an upright box, and treats h_{2D} as the tight projection of the eight 3D corners. Second, PCDepth couples depth to the predicted 3D size and yaw, so depth accuracy depends on the accuracy of these attributes; we discuss this dependency in Sec. 3 and address it through GeoAlign, but residual attribute errors (*e.g.*, on heavily occluded or distant objects with weak visual cues) still propagate to depth. Third, our evaluation focuses on outdoor driving benchmarks (KITTI, nuScenes, Waymo, and Omni3D), and the behavior of MonoPCA under significantly different camera setups has not been studied in this work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not contain formal theoretical results. The geometric derivation of PCDepth in Sec. 3 relies on standard perspective projection.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the information needed to reproduce our results. The architecture of MonoPCA, including PCDepth, MoB, and HFA, is fully described in Sec. 3 and Appendix A, with all training losses and loss weights specified. Implementation details (ResNet-50 backbone, EVA-CLIP as the VFM, MonoCoP [94] as the base detector, AdamW optimizer with the learning rate schedule, batch size, and hardware setup) are reported in Sec. 4 and Appendix A. All experiments use publicly available benchmarks (KITTI, nuScenes, Waymo, and Omni3D) under their standard splits, with dataset sources cited. The code and trained checkpoints are publicly available at `https://github.com/alanzhangcs/MonoPCA`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We release a public repository (https://github.com/alanzhangcs/MonoPCA) containing the full source code of MonoPCA, configuration files, training scripts, and instructions for environment setup. All datasets used (KITTI, nuScenes, Waymo, and Omni3D) are publicly available under their respective licenses, and we follow their standard preprocessing protocols, with details described in Sec. 4 and Appendix A. The repository includes step-by-step commands to reproduce the main experimental results reported in the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental details are reported in Sec. 4 and Appendix A, including dataset splits for KITTI, nuScenes, Waymo, and Omni3D, evaluation metrics and IoU

thresholds, the optimizer (AdamW) with its learning rate, weight decay, and decay schedule, the batch size, the number of training epochs, the loss weights for all training objectives, and the hardware setup. Hyperparameters follow the conventions of the base detector [94] and are not separately tuned for our method.

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars. Following the standard evaluation protocol of Mono3D on KITTI, nuScenes, Waymo, and Omni3D [89, 62, 94, 3], results are obtained from a single training run per configuration, since each run requires substantial compute (eight NVIDIA A100 GPUs). Repeated runs across all benchmarks and ablation settings would be computationally prohibitive. To compensate, we evaluate MonoPCA on four diverse benchmarks and observe consistent improvements over strong baselines across datasets, difficulty levels, and detector architectures (Fig. 7), which we believe supports the statistical reliability of our reported gains.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments are conducted on a single node with eight NVIDIA A100 GPUs (80GB) and a batch size of 16, as reported in Appendix A. A typical full training run on KITTI takes approximately 12 hours, while training on the larger Waymo and Omni3D benchmarks takes one to two days. Inference cost is reported in Tab. 3, where MonoPCA runs at 62.5 GFLOPs with 42.0M parameters, slightly more efficient than the MonoCoP [94] baseline. We acknowledge that the full research project, including preliminary explorations and failed designs, required compute beyond the final reported experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics, and our research fully conforms to its guidelines. The work uses publicly available autonomous driving benchmarks (KITTI, nuScenes, Waymo, and Omni3D) under their respective licenses, does not involve human subjects or private data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: MonoPCA improves monocular 3D object detection, which has direct positive applications in autonomous driving, robotics, and augmented reality, where reliable depth perception from a single camera reduces dependency on expensive LiDAR sensors and broadens accessibility to safety-critical perception. As with any 3D detection technology, potential negative impacts include misuse in unauthorized surveillance and the risk of perception errors in deployed autonomous systems leading to safety hazards. We emphasize that MonoPCA is intended as research progress on geometric depth modeling rather than a deployment-ready system, and any real-world use should be accompanied by rigorous validation, redundant sensing modalities, and adherence to relevant safety standards.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work releases a monocular 3D object detector for outdoor driving scenes. The model is not a generative system, does not produce or scrape data, and does not pose a high risk of misuse such as disinformation or harmful content generation. The released code and checkpoints are intended for research purposes on standard public benchmarks, and therefore no additional safeguards beyond those discussed in the broader-impact statement are required.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing assets used in this work are properly cited and used in accordance with their licenses. We use four public benchmarks: KITTI [18] (CC BY-NC-SA 3.0), nuScenes [6] (CC BY-NC-SA 4.0), Waymo Open Dataset [75] (Waymo Dataset License Agreement for non-commercial use), and Omni3D [3] (CC BY-NC 4.0), all employed under their non-commercial research terms. Our implementation builds on MonoCoP [94] and adopts CLIP [76] as the Vision Foundation Model; both are released for research use and properly credited in Sec. 4 and Appendix A.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new asset introduced in this work is the source code of MonoPCA, including the implementation of PCDepth, MoB, and HFA. We release a public repository (https://github.com/alanzhangcs/MonoPCA) containing the source code, configuration files, training and evaluation scripts, environment setup instructions, and a README documenting usage, together with the trained checkpoints.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects. All experiments use existing public benchmarks for monocular 3D object detection.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as part of the core methodology of this research. The proposed MonoPCA, including PCDepth and GeoAlign, does not involve any LLM component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.