DEVIANT: Depth EquiVarIAnt NeTwork for Monocular 3D Object Detection

Abhinav Kumar[1], Garrick Brazil[2], Enrique Corona[3], Armin Parchami[3], and Xiaoming Liu[1]

[1] Michigan State University
[2] Meta AI
[3] Ford Motor Company
[1]{kumarab6, liuxm}@msu.edu,[2]brazilga@fb.com,[3]{ecoron18, mparcham}@ford.com
https://github.com/abhi1kumar/DEVIANT

Abstract. Modern neural networks use building blocks such as convolutions that are equivariant to arbitrary 2D translations. However, these vanilla blocks are not equivariant to arbitrary 3D translations in the projective manifold. Even then, all monocular 3D detectors use vanilla blocks to obtain the 3D coordinates, a task for which the vanilla blocks are not designed for. This paper takes the first step towards convolutions equivariant to arbitrary 3D translations in the projective manifold. Since the depth is the hardest to estimate for monocular detection, this paper proposes Depth EquiVarIAnt NeTwork (DEVIANT) built with existing scale equivariant steerable blocks. As a result, DEVIANT is equivariant to the depth translations in the projective manifold whereas vanilla networks are not. The additional depth equivariance forces the DEVIANT to learn consistent depth estimates, and therefore, DEVIANT achieves state-of-the-art monocular 3D detection results on KITTI and Waymo datasets in the image-only category and performs competitively to methods using extra information. Moreover, DEVIANT works better than vanilla networks in cross-dataset evaluation.

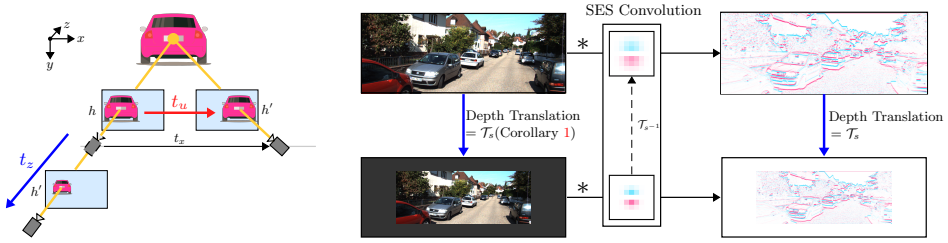
Keywords: Equivariance, Projective manifold, Monocular 3D detection

1 Introduction

Monocular 3D object detection is a fundamental task in computer vision, where the task is to infer 3D information including depth from a single monocular image. It has applications in augmented reality [2], gaming [63], robotics [65], and more recently in autonomous driving [4,68] as a fallback solution for LiDAR.

Most of the monocular 3D methods attach extra heads to the 2D Faster-RCNN [64] or CenterNet [102] for 3D detections. Some change architectures [42, 45, 76] or losses [4, 13]. Others incorporate augmentation [71], or confidence [5,45]. Recent ones use in-network ensembles [49,100] for better depth estimation.

Most of these methods use vanilla blocks such as convolutions that are *equivariant* to arbitrary 2D translations [6,61]. In other words, whenever we shift the ego camera in 2D (See t_u of Fig. 1), the new image (projection) is a translation of



(a) Idea. (b) Depth Equivariance.

Fig. 1: **(a) Idea.** Vanilla CNN is equivariant to projected 2D translations t_u, t_v of the ego camera. The ego camera moves in 3D in driving scenes which breaks this assumption. We propose DEVIANT which is additionally equivariant to depth translations t_z in the projective manifold. **(b) Depth Equivariance**. DEVIANT enforces additional consistency among the feature maps of an image and its transformation caused by the ego depth translation. $\mathcal{T}_s$ = scale transformation, $*$ = vanilla convolution.

Table 1: **Equivariance comparisons**. [Key: Proj.= Projected, ax= axis]

Translation →	3D			Proj. 2D	
	$x-$ax (t_x)	$y-$ax (t_y)	$z-$ax (t_z)	u-ax (t_u)	v-ax (t_v)
Vanilla CNN	–	–	–	✓	✓
Log-polar [106]	–	–	✓	–	–
DEVIANT	–	–	✓	✓	✓
Ideal	✓	✓	✓	–	–

the original image, and therefore, these methods output a translated feature map. However, in general, the camera moves in depth in driving scenes instead of 2D (See t_z of Fig. 1). So, the new image is not a translation of the original input image due to the projective transform. Thus, using vanilla blocks in monocular methods is a mismatch between the assumptions and the regime where these blocks operate. Additionally, there is a huge generalization gap between training and validation for monocular 3D detection (See Tab. 14 in the supplementary). Modeling translation equivariance in the correct manifold improves generalization for tasks in spherical [15] and hyperbolic [26] manifolds. Monocular detection involves processing pixels (3D point projections) to obtain the 3D information, and is thus a task in the projective manifold. Moreover, the depth in monocular detection is ill-defined [76], and thus, the hardest to estimate [53]. Hence, using building blocks *equivariant to depth translations in the projective manifold* is a natural choice for improving generalization and is also at the core of this work (See Appendix A1.8).

Recent monocular methods use flips [4], scale [49,71], mosaic [3,77] or copy-paste [43] augmentation, depth-aware convolution [4], or geometry [47,49,67,99] to improve generalization. Although all these methods improve performance, a major issue is that their backbones are not designed for the projective world. This results in the depth estimation going haywire with a slight ego movement [103]. Moreover, data augmentation, *e.g.*, flips, scales, mosaic, copy-paste, is not only limited for the projective tasks, but also does not guarantee desired behavior [25].

To address the mismatch between assumptions and the operating regime of the vanilla blocks and improve generalization, we take the first step towards convolutions equivariant to arbitrary 3D translations in the projective mani-

Table 2: Equivariances known in the literature.

Transformation → / **Manifold ↓**	Translation	Rotation	Scale	Flips	Learned
Euclidean	Vanilla CNN [40]	Polar, Steerable [91]	Log-polar [31], Steerable [29]	ChiralNets [96]	Transformers [21]
Spherical	Spherical CNN [15]	–	–	–	–
Hyperbolic	Hyperbolic CNN [26]	–	–	–	–
Projective	Monocular Detector	–	–	–	–

fold. We propose Depth EquiVarIAnt NeTwork (DEVIANT) which is additionally equivariant to depth translations in the projective manifold as shown in Tab. 1. Building upon the classic result from [30], we simplify it under reasonable assumptions about the camera movement in autonomous driving to get scale transformations. The scale equivariant blocks are well-known in the literature [29, 32, 74, 104], and consequently, we replace the vanilla blocks in the backbone with their scale equivariant steerable counterparts [74] to additionally embed equivariance to depth translations in the projective manifold. Hence, DEVIANT learns consistent depth estimates and improves monocular detection.

In summary, the main contributions of this work include:

- We study the modeling error in monocular 3D detection and propose depth equivariant networks built with scale equivariant steerable blocks as a solution.
- We achieve state-of-the-art (SOTA) monocular 3D object detection results on the KITTI and Waymo datasets in the image-only category and perform competitively to methods which use extra information.
- We experimentally show that DEVIANT works better in cross-dataset evaluation suggesting better generalization than vanilla CNN backbones.

2 Literature Review

Equivariant Neural Networks. The success of convolutions in CNN has led people to look for their generalizations [17, 87]. Convolution is the unique solution to 2D translation equivariance in the Euclidean manifold [6, 7, 61]. Thus, convolution in CNN is a prior in the Euclidean manifold. Several works explore other group actions in the Euclidean manifold such as 2D rotations [16, 19, 55, 88], scale [34, 54], flips [96], or their combinations [81, 91]. Some consider 3D translations [90] and rotations [78]. Few [21, 89, 101] attempt learning the equivariance from the data, but such methods have significantly higher data requirements [90]. Others change the manifold to spherical [15], hyperbolic [26], graphs [56], or arbitrary manifolds [33]. Monocular 3D detection involves operations on pixels which are projections of 3D point and thus, works in a different manifold namely projective manifold. Tab. 2 summarizes all these equivariances known thus far.

Scale Equivariant Networks. Scale equivariance in the Euclidean manifold is more challenging than the rotations because of its acyclic and unbounded nature [61]. There are two major lines of work for scale equivariant networks. The first [22, 31] infers the global scale using log-polar transform [106], while the other infers the scale locally by convolving with multiple scales of images [34] or

filters [94]. Several works [29, 32, 74, 104] extend the local idea, using steerable filters [24]. Another work [92] constructs filters for integer scaling. We compare the two kinds of scale equivariant convolutions on the monocular 3D detection task and show that steerable convolutions are better suited to embed depth (scale) equivariance. Scale equivariant networks have been used for classification [22, 29, 74], 2D tracking [73] and 3D object classification [22]. We are the first to use scale equivariant networks for monocular 3D detection.

3D Object Detection. Accurate 3D object detection uses sparse data from LiDARs [66], which are expensive and do not work well in severe weather [76] and glassy environments. Hence, several works have been on monocular camera-based 3D object detection, which is simplistic but has scale/depth ambiguity [76]. Earlier approaches [11, 23, 59, 60] use hand-crafted features, while the recent ones use deep learning. Some change architectures [42, 45, 46, 76] or losses [4, 13]. Some use scale [49, 71], mosaic [77] or copy-paste [43] augmentation. Others incorporate depth in convolution [4, 20], or confidence [5, 37, 45]. More recent ones use in-network ensembles to predict the depth deterministically [100] or probabilistically [49]. A few use temporal cues [5], NMS [36], or corrected camera extrinsics [103] in the training pipeline. Some also use CAD models [10, 48] or LiDAR [62] in training. Another line of work called Pseudo-LiDAR [50, 52, 57, 69, 83] estimates the depth first, and then uses a point cloud-based 3D object detector. We refer to [51] for a detailed survey. Our work is the first to use scale equivariant blocks in the backbone for monocular 3D detection.

3 Background

We first provide the necessary definitions which are used throughout this paper. These are not our contributions and can be found in the literature [8, 30, 90].

Equivariance. Consider a group of transformations G, whose individual members are g. Assume Φ denote the mapping of the inputs h to the outputs y. Let the inputs and outputs undergo the transformation $\mathcal{T}_g^h$ and $\mathcal{T}_g^y$ respectively. Then, the mapping Φ is equivariant to the group G [90] if $\Phi(\mathcal{T}_g^h h) = \mathcal{T}_g^y(\Phi h), \forall\ g \in G$. Thus, equivariance provides an explicit relationship between input transformations and feature-space transformations at each layer of the neural network [90], and intuitively makes the learning easier. The mapping Φ is the vanilla convolution when the $\mathcal{T}_g^h = \mathcal{T}_g^y = \mathcal{T}_\mathbf{t}$ where $\mathcal{T}_\mathbf{t}$ denotes the translation $\mathbf{t}$ on the discrete grid [6, 7, 61]. These vanilla convolution introduce weight-tying [40] in fully connected neural networks resulting in a greater generalization. A special case of equivariance is the invariance [90] which is given by $\Phi(\mathcal{T}_g^h h) = \Phi h, \forall\ g \in G$.

Projective Transformations. Our idea is to use equivariance to depth translations in the projective manifold since the monocular detection task belongs to this manifold. A natural question to ask is whether such equivariants exist in the projective manifold. [8] answers this question in negative, and says that such equivariants do not exist in general. However, such equivariants exist for special classes, such as planes. An intuitive way to understand this is to infer the rotations and translations by looking at the two projections (images). For

example, the result of [8] makes sense if we consider a car with very different front and back sides as in Fig. 6. A 180° ego rotation around the car means the projections (images) are its front and the back sides, which are different. Thus, we can not infer the translations and rotations from these two projections. Based on this result, we stick with **locally** planar objects *i.e.* we assume that a 3D object is made of several *patch planes*. (See last row of Fig. 2b as an example). It is important to stress that we do **NOT** assume that the 3D object such as car is planar. The local planarity also agrees with the property of manifolds that manifolds locally resemble n-dimensional Euclidean space and because the projective transform maps planes to planes, the patch planes in 3D are also locally planar. We show a sample planar patch and the 3D object in Fig. 5 in the appendix.

Planarity and Projective Transformation. Example 13.2 from [30] links the planarity and projective transformations. Although their result is for stereo with two different cameras $(\mathbf{K}, \mathbf{K}')$, we substitute $\mathbf{K}=\mathbf{K}'$ to get Theorem 1.

Theorem 1. *[30] Consider a 3D point lying on a patch plane $mx+ny+oz+p=0$, and observed by an ego camera in a pinhole setup to give an image h. Let $\mathbf{t}=(t_x,t_y,t_z)$ and $\mathbf{R}=[r_{ij}]_{3\times 3}$ denote a translation and rotation of the ego camera respectively. Observing the same 3D point from a new camera position leads to an image h'. Then, the image h is related to the image h' by the projective transformation*

$$\mathcal{T}: h(u-u_0, v-v_0) = \tag{1}$$
$$h'\left(\frac{f\left(r_{11}+\bar{t}_x\frac{m}{p}\right)(u-u_0)+\left(r_{21}+\bar{t}_x\frac{n}{p}\right)(v-v_0)+\left(r_{31}+\bar{t}_x\frac{o}{p}\right)f}{\left(r_{13}+\bar{t}_z\frac{m}{p}\right)(u-u_0)+\left(r_{23}+\bar{t}_z\frac{n}{p}\right)(v-v_0)+\left(r_{33}+\bar{t}_z\frac{o}{p}\right)f},\right.$$
$$\left.\frac{f\left(r_{12}+\bar{t}_y\frac{m}{p}\right)(u-u_0)+\left(r_{22}+\bar{t}_y\frac{n}{p}\right)(v-v_0)+\left(r_{32}+\bar{t}_y\frac{o}{p}\right)f}{\left(r_{13}+\bar{t}_z\frac{m}{p}\right)(u-u_0)+\left(r_{23}+\bar{t}_z\frac{n}{p}\right)(v-v_0)+\left(r_{33}+\bar{t}_z\frac{o}{p}\right)f}\right),$$

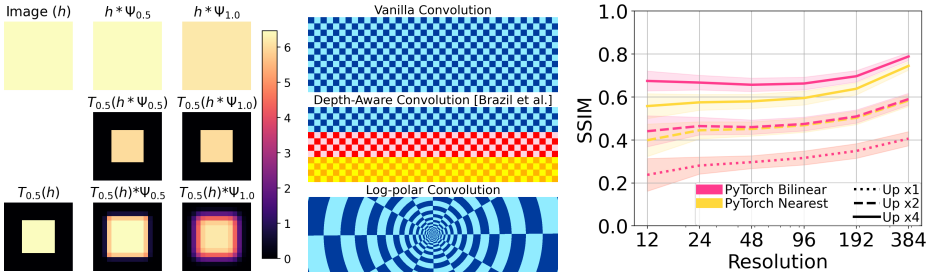
where f and (u_0, v_0) denote the focal length and principal point of the ego camera, and $(\bar{t}_x, \bar{t}_y, \bar{t}_z) = \mathbf{R}^T\mathbf{t}$.

4 Depth Equivariant Backbone

The projective transformation in Eq. (1) from [30] is complicated and also involves rotations, and we do not know which convolution obeys this projective transformation. Hence, we simplify Eq. (1) under reasonable assumptions to obtain a familiar transformation for which the *convolution* is known.

Corollary 1. *When the ego camera translates in depth without rotations ($\mathbf{R}=\mathbf{I}$), and the patch plane is "approximately" parallel to the image plane, the image h locally is a scaled version of the second image h' independent of focal length, i.e.*

$$\mathcal{T}_s: h(u-u_0, v-v_0) \approx h'\left(\frac{u-u_0}{1+t_z\frac{o}{p}}, \frac{v-v_0}{1+t_z\frac{o}{p}}\right). \tag{2}$$



(a) SES Convolution Output. (b) Receptive fields. (c) Log-polar SSIM.

Fig. 2: **(a) Scale Equivariance**. We apply SES convolution [74] with two scales on a single channel toy image h. **(b) Receptive fields** of convolutions in the Euclidean manifold. Colors represent different weights, while shades represent the same weight. **(c) Impact of discretization on log-polar convolution.** SSIM is very low at small resolutions and is not 1 even after upscaling by 4. [Key: Up= Upscaling]

where f and (u_0, v_0) denote the focal length and principal point of the ego camera, and t_z denotes the ego translation.

See Appendix A1.6 for the detailed explanation of Corollary 1. Corollary 1 says

$$\mathcal{T}_s : h(u - u_0, v - v_0) \approx h'\left(\frac{u - u_0}{s}, \frac{v - v_0}{s}\right), \tag{3}$$

where, $s = 1 + t_z \frac{o}{p}$ denotes the scale and $\mathcal{T}_s$ denotes the scale transformation. The scale $s < 1$ suggests downscaling, while $s > 1$ suggests upscaling. Corollary 1 shows that the transformation $\mathcal{T}_s$ is independent of the focal length and that scale is a linear function of the depth translation. Hence, the depth translation in the projective manifold induces scale transformation and thus, the depth equivariance in the projective manifold is the scale equivariance in the Euclidean manifold. Mathematically, the desired equivariance is $[\mathcal{T}_s(h) * \Psi] = \mathcal{T}_s\,[h * \Psi_{s^{-1}}]$, where Ψ denotes the filter (See Appendix A1.7). As CNN is not a scale equivariant (SE) architecture [74], we aim to get SE backbone which makes the architecture equivariant to depth translations in the projective manifold. The scale transformation is a familiar transformation and SE convolutions are well known [29, 32, 74, 104].

Scale Equivariant Steerable (SES) Blocks. We use the existing SES blocks [73, 74] to construct our Depth EquiVarIAnt NeTwork (DEVIANT) backbone. As [73] does not construct SE-DLA-34 backbones, we construct our DEVIANT backbone as follows. We replace the vanilla convolutions by the SES convolutions [73] with the basis as Hermite polynomials. SES convolutions result in multi-scale representation of an input tensor. As a result, their output is five-dimensional instead of four-dimensional. Thus, we replace the 2D pools and batch norm (BN) by 3D pools and 3D BN respectively. The Scale-Projection layer [74] carries a max over the extra (scale) dimension to project five-dimensional tensors to four dimensions (See Fig. 9 in the supplementary). Ablation in Sec. 5.2 confirms that BN and Pool (BNP) should also be SE for the best performance.

The SES convolutions [29, 74, 104] are based on steerable-filters [24]. Steerable approaches [29] first pre-calculate the non-trainable multi-scale basis in the Euclidean manifold and then build filters by the linear combinations of the trainable weights $\mathbf{w}$ (See Fig. 9). The number of trainable weights $\mathbf{w}$ equals the number of filters at one particular scale. The linear combination of multi-scale basis ensures that the filters are also multi-scale. Thus, SES blocks bypass grid conversion and do not suffer from sampling effects.

We show the convolution of toy image h with a SES convolution in Fig. 2a. Let Ψ_s denote the filter at scale s. The convolution between downscaled image and filter $\mathcal{T}_{0.5}(h) * \Psi_{0.5}$ matches the downscaled version of original image convolved with upscaled filter $\mathcal{T}_{0.5}(h * \Psi_{1.0})$. Fig. 2a (right column) shows that the output of a CNN exhibits aliasing in general and is therefore, not scale equivariant.

Log-polar Convolution: Impact of Discretization. An alternate way to convert the depth translation t_z of Eq. (2) to shift is by converting the images to log-polar space [106] around the principal point (u_0, v_0), as

$$h(\ln r, \theta) \approx h'\left(\ln r - \ln\left(1 + t_z \frac{o}{p}\right), \quad \theta\right), \tag{4}$$

with $r = \sqrt{(u-u_0)^2 + (v-v_0)^2}$, and $\theta = \tan^{-1}\left(\frac{v-v_0}{u-u_0}\right)$. The log-polar transformation converts the scale to translation, so using convolution in the log-polar space is equivariant to the logarithm of the depth translation t_z. We show the receptive field of log-polar convolution in Fig. 2b. The log-polar convolution uses a smaller receptive field for objects closer to the principal point, while a larger field away from the principal point. We implemented log-polar convolution and found that its performance (See Tab. 11) is not acceptable, consistent with [74]. We attribute this behavior to the discretization of pixels and loss of 2D translation equivariance. Eq. (4) is perfectly valid in the continuous world (Note the use of parentheses instead of square brackets in Eq. (4)). However, pixels reside on discrete grids, which gives rise to sampling errors [38]. We discuss the impact of discretization on log-polar convolution in Sec. 5.2 and show it in Fig. 2c. Hence, we do not use log-polar convolution for the DEVIANT backbone.

Comparison of Equivariances for Monocular 3D Detection. We now compare equivariances for monocular 3D detection task. An ideal monocular detector should be equivariant to arbitrary 3D translations (t_x, t_y, t_z). However, most monocular detectors [36, 49] estimate 2D projections of 3D centers and the depth, which they back-project in 3D world via known camera intrinsics. Thus, a good enough detector shall be equivariant to 2D translations (t_u, t_v) for projected centers as well as equivariant to depth translations (t_z).

Existing detector backbones [36, 49] are only equivariant to 2D translations as they use vanilla convolutions that produce 4D feature maps. Log-polar backbones is equivariant to logarithm of depth translations but not to 2D translations. DEVIANT uses SES convolutions to produce 5D feature maps. The extra dimension in 5D feature map captures the changes in scale (for depth), while these feature maps individually are equivariant to 2D translations (for projected centers). Hence, DEVIANT augments the 2D translation equivariance (t_u, t_v)

of the projected centers with the depth translation equivariance. We emphasize that although DEVIANT is **not** equivariant to arbitrary 3D translations in the projective manifold, DEVIANT **does** provide the equivariance to depth translations (t_z) and is thus a first step towards the ideal equivariance. Our experiments (Sec. 5) show that even this additional equivariance benefits monocular 3D detection task. This is expected because depth is the hardest parameter to estimate [53]. Tab. 1 summarizes these equivariances. Moreover, Tab. 10 empirically shows that 2D detection does not suffer and therefore, confirms that DEVIANT indeed augments the 2D equivariance with the depth equivariance. An idea similar to DEVIANT is the optical expansion [95] which augments optical flow with the scale information and benefits depth estimation.

5 Experiments

Our experiments use the KITTI [28], Waymo [75] and nuScenes datasets [9]. We modify the publicly-available PyTorch [58] code of GUP Net [49] and use the GUP Net model as our baseline. For DEVIANT, we keep the number of scales as three [73]. DEVIANT takes 8.5 hours for training and 0.04s per image for inference on a single A100 GPU. See Appendix A2.2 for more details.

Evaluation Metrics. KITTI evaluates on three object categories: Easy, Moderate and Hard. It assigns each object to a category based on its occlusion, truncation, and height in the image space. KITTI uses $\text{AP}_{\text{3D}|R_{40}}$ percentage metric on the Moderate category to benchmark models [28] following [68, 70].

Waymo evaluates on two object levels: Level_1 and Level_2. It assigns each object to a level based on the number of LiDAR points included in its 3D box. Waymo uses APH_{3D} percentage metric which is the incorporation of heading information in AP_{3D} to benchmark models. It also provides evaluation at three distances $[0, 30)$, $[30, 50)$ and $[50, \infty)$ meters.

Data Splits. We use the following splits of the KITTI,Waymo and nuScenes:

- *KITTI Test (Full) split*: Official KITTI 3D benchmark [1] consists of 7,481 training and 7,518 testing images [28].
- *KITTI Val split*: It partitions the 7,481 training images into 3,712 training and 3,769 validation images [12].
- *Waymo Val split*: This split [62,80] contains 52,386 training and 39,848 validation images from the front camera. We construct its training set by sampling every third frame from the training sequences as in [62, 80].
- *nuScenes Val split:* It consists of 28,130 training and 6,019 validation images from the front camera [9]. We use this split for evaluation [67].

5.1 KITTI Test Monocular 3D Detection

Cars. Tab. 3 lists out the results of monocular 3D detection and BEV evaluation on KITTI Test cars. Tab. 3 results show that DEVIANT outperforms the GUP Net and several other SOTA methods on both tasks. Except DD3D [57] and

Table 3: **Results on KITTI Test cars** at $\text{IoU}_{3D} \geq 0.7$. Previous results are from the leader-board or papers. We show 3 methods in each Extra category and 6 methods in the image-only category. [Key: **Best**, **Second Best**]

Method	Extra	$AP_{3D\|R_{40}}$[%](↑)			$AP_{BEV\|R_{40}}$[%](↑)		
		Easy	Mod	Hard	Easy	Mod	Hard
AutoShape [48]	CAD	22.47	14.17	11.36	30.66	20.08	15.59
PCT [80]	Depth	21.00	13.37	11.31	29.65	19.03	15.92
DFR-Net [105]	Depth	19.40	13.63	10.35	28.17	19.17	14.84
MonoDistill [14]	Depth	22.97	16.03	13.60	31.87	22.59	19.72
PatchNet-C [69]	LiDAR	22.40	12.53	10.60	–	–	–
CaDDN [62]	LiDAR	19.17	13.41	11.46	27.94	18.91	17.19
DD3D [57]	LiDAR	23.22	16.34	14.20	30.98	22.56	20.03
MonoEF [103]	Odometry	21.29	13.87	11.71	29.03	19.70	17.26
Kinematic [5]	Video	19.07	12.72	9.17	26.69	17.52	13.10
GrooMeD-NMS [36]	–	18.10	12.32	9.65	26.19	18.27	14.05
MonoRCNN [67]	–	18.36	12.65	10.03	25.48	18.11	14.10
MonoDIS-M [68]	–	16.54	12.97	11.04	24.45	19.25	16.87
Ground-Aware [47]	–	**21.65**	13.25	9.91	**29.81**	17.98	13.08
MonoFlex [100]	–	19.94	13.89	**12.07**	28.23	**19.75**	**16.89**
GUP Net [49]	–	20.11	**14.20**	11.77	–	–	–
DEVIANT (Ours)	–	**21.88**	**14.46**	**11.89**	**29.65**	**20.44**	**17.43**

Table 4: **Results on KITTI Test cyclists and pedestrians** (Cyc/Ped) at $\text{IoU}_{3D} \geq 0.5$. Previous results are from the leader-board or papers. [Key: **Best**, **Second Best**]

Method	Extra	Cyc $AP_{3D\|R_{40}}$[%](↑)			Ped $AP_{3D\|R_{40}}$[%](↑)		
		Easy	Mod	Hard	Easy	Mod	Hard
DDMP-3D [79]	Depth	4.18	2.50	2.32	4.93	3.55	3.01
DFR-Net [105]	Depth	5.69	3.58	3.10	6.09	3.62	3.39
MonoDistill [14]	Depth	5.53	2.81	2.40	12.79	8.17	7.45
CaDDN [62]	LiDAR	7.00	3.41	3.30	12.87	8.14	6.76
DD3D [57]	LiDAR	2.39	1.52	1.31	13.91	9.30	8.05
MonoEF [103]	Odometry	1.80	0.92	0.71	4.27	2.79	2.21
MonoDIS-M [68]	–	1.17	0.54	0.48	7.79	5.14	4.42
MonoFlex [100]	–	3.39	2.10	1.67	11.89	8.16	6.81
GUP Net [49]	–	**4.18**	**2.65**	**2.09**	**14.72**	**9.53**	**7.87**
DEVIANT (Ours)	–	**5.05**	**3.13**	**2.59**	**13.43**	**8.65**	**7.69**

MonoDistill [14], DEVIANT, an image-based method, also outperforms other methods that use extra information.

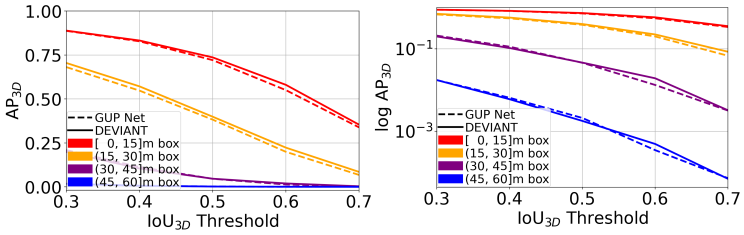
Cyclists and Pedestrians. Tab. 4 lists out the results of monocular 3D detection on KITTI Test Cyclist and Pedestrians. The results show that DEVIANT achieves SOTA results in the image-only category on the challenging Cyclists, and is competitive on Pedestrians.

5.2 KITTI Val Monocular 3D Detection

Cars. Tab. 5 summarizes the results of monocular 3D detection and BEV evaluation on KITTI Val split at two IoU_{3D} thresholds of 0.7 and 0.5 [13,36]. We report the **median** model over 5 runs. The results show that DEVIANT outperforms the GUP Net [49] baseline by a significant margin. The biggest improvements shows up on the Easy set. Significant improvements are also on the Moderate and Hard sets. Interestingly, DEVIANT also outperforms DD3D [57] by a large margin when the large-dataset pretraining is not done (denoted by DD3D^{-}).

Table 5: **Results on KITTI Val cars**. Comparison with bigger CNN backbones in Tab. 16. [Key: **Best**, **Second Best**, $^{-}$= No pretrain]

Method	Extra	$IoU_{3D} \geq 0.7$						$IoU_{3D} \geq 0.5$					
		$AP_{3D\|R_{40}}$[%](↑)			$AP_{BEV\|R_{40}}$[%](↑)			$AP_{3D\|R_{40}}$[%](↑)			$AP_{BEV\|R_{40}}$[%](↑)		
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
DDMP-3D [79]	Depth	28.12	20.39	16.34	–	–	–	–	–	–	–	–	–
PCT [80]	Depth	38.39	27.53	24.44	47.16	34.65	28.47	–	–	–	–	–	–
MonoDistill [14]	Depth	24.31	18.47	15.76	33.09	25.40	22.16	65.69	49.35	43.49	71.45	53.11	46.94
CaDDN [62]	LiDAR	23.57	16.31	13.84	–	–	–	–	–	–	–	–	–
PatchNet-C [69]	LiDAR	24.51	17.03	13.25	–	–	–	–	–	–	–	–	–
DD3D (DLA34) [57]	LiDAR	–	–	–	33.5	26.0	22.6	–	–	–	–	–	–
DD3D^{-}(DLA34) [57]	LiDAR	–	–	–	26.8	20.2	16.7	–	–	–	–	–	–
MonoEF [103]	Odometry	18.26	16.30	15.24	26.07	25.21	21.61	57.98	51.80	49.34	63.40	61.13	53.22
Kinematic [5]	Video	19.76	14.10	10.47	27.83	19.72	15.10	55.44	39.47	31.26	61.79	44.68	34.56
MonoRCNN [67]	–	16.61	13.19	10.65	25.29	19.22	15.30	–	–	–	–	–	–
MonoDLE [53]	–	17.45	13.66	11.68	24.97	19.33	17.01	55.41	43.42	37.81	60.73	46.87	41.89
GrooMeD-NMS [36]	–	19.67	14.32	11.27	27.38	19.75	15.92	55.62	41.07	32.89	61.83	44.98	36.29
Ground-Aware [47]	–	23.63	16.16	12.06	–	–	–	**60.92**	42.18	32.02	–	–	–
MonoFlex [100]	–	**23.64**	**17.51**	**14.83**	–	–	–	–	–	–	–	–	–
GUP Net (Reported) [49]	–	22.76	16.46	13.72	**31.07**	**22.94**	**19.75**	57.62	42.33	37.59	61.78	47.06	40.88
GUP Net (Retrained) [49]	–	21.10	15.48	12.88	28.58	20.92	17.83	58.95	**43.99**	**38.07**	**64.60**	**47.76**	**42.97**
DEVIANT (Ours)	–	**24.63**	**16.54**	**14.52**	**32.60**	**23.04**	**19.99**	**61.00**	**46.00**	**40.18**	**65.28**	**49.63**	**43.50**



(a) Linear Scale (b) Log Scale

Fig. 3: **AP_{3D} at different depths and IoU_{3D} thresholds** on KITTI Val Split.

Table 6: **Cross-dataset evaluation** of the KITTI Val model on KITTI Val and nuScenes frontal Val cars with depth MAE (↓). [Key: **Best**, **Second Best**]

Method	KITTI Val				nuScenes frontal Val			
	0−20	20−40	40−∞	All	0−20	20−40	40−∞	All
M3D-RPN [4]	0.56	1.33	2.73	1.26	0.94	3.06	10.36	2.67
MonoRCNN [67]	0.46	1.27	2.59	1.14	0.94	2.84	8.65	2.39
GUP Net [49]	**0.45**	**1.10**	**1.85**	**0.89**	**0.82**	**1.70**	**6.20**	**1.45**
DEVIANT	**0.40**	**1.09**	**1.80**	**0.87**	**0.76**	**1.60**	**4.50**	**1.26**

AP_{3D} at different depths and IoU_{3D} thresholds. We next compare the AP_{3D} of DEVIANT and GUP Net in Fig. 3 at different distances in meters and IoU_{3D} matching criteria of $0.3 \rightarrow 0.7$ as in [36]. Fig. 3 shows that DEVIANT is effective over GUP Net [49] at all depths and higher IoU_{3D} thresholds.

Cross-Dataset Evaluation. Tab. 6 shows the result of our KITTI Val model on the KITTI Val and nuScenes [9] frontal Val images, using mean absolute error (MAE) of the depth of the boxes [67]. More details are in Appendix A3.1. DEVIANT outperforms GUP Net on most of the metrics on both the datasets, which confirms that DEVIANT generalizes better than CNNs. DEVIANT per-

Table 7: **Scale Augmentation vs Scale Equivariance** on KITTI Val cars. [Key: **Best**, Eqv= Equivariance, Aug= Augmentation]

Method	Scale Eqv	Scale Aug	$IoU_{3D} \geq 0.7$						$IoU_{3D} \geq 0.5$					
			$AP_{3D\|R_{40}}$[%](↑)			$AP_{BEV\|R_{40}}$[%](↑)			$AP_{3D\|R_{40}}$[%](↑)			$AP_{BEV\|R_{40}}$[%](↑)		
			Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
GUP Net [49]			20.82	14.15	12.44	29.93	20.90	17.87	**62.37**	44.40	39.61	**66.81**	48.09	43.14
		✓	21.10	15.48	12.88	28.58	20.92	17.83	58.95	43.99	38.07	64.60	47.76	42.97
DEVIANT	✓		21.33	14.77	12.57	28.79	20.28	17.59	59.31	43.25	37.64	63.94	47.02	41.12
	✓	✓	**24.63**	**16.54**	**14.52**	**32.60**	**23.04**	**19.99**	61.00	**46.00**	**40.18**	65.28	**49.63**	**43.50**

Table 8: **Comparison of Equivariant Architectures** on KITTI Val cars. [Key: **Best**, Eqv= Equivariance, †= Retrained]

Method	Eqv	$IoU_{3D} \geq 0.7$						$IoU_{3D} \geq 0.5$					
		$AP_{3D\|R_{40}}$[%](↑)			$AP_{BEV\|R_{40}}$[%](↑)			$AP_{3D\|R_{40}}$[%](↑)			$AP_{BEV\|R_{40}}$[%](↑)		
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
DETR3D† [85]	Learned	1.94	1.26	1.09	4.41	3.06	2.79	20.09	13.80	12.78	26.51	18.49	17.36
GUP Net [49]	2D	21.10	15.48	12.88	28.58	20.92	17.83	58.95	43.99	38.07	64.60	47.76	42.97
DEVIANT	2D+Depth	**24.63**	**16.54**	**14.52**	**32.60**	**23.04**	**19.99**	**61.00**	**46.00**	**40.18**	**65.28**	**49.63**	**43.50**

Table 9: **Comparison with Dilated Convolution** on KITTI Val cars. [Key: **Best**]

Method	Extra	$IoU_{3D} \geq 0.7$						$IoU_{3D} \geq 0.5$					
		$AP_{3D\|R_{40}}$[%](↑)			$AP_{BEV\|R_{40}}$[%](↑)			$AP_{3D\|R_{40}}$[%](↑)			$AP_{BEV\|R_{40}}$[%](↑)		
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
D4LCN [20]	Depth	22.32	16.20	12.30	31.53	22.58	17.87	–	–	–	–	–	–
DCNN [97]	–	21.66	15.49	12.90	30.22	22.06	19.01	57.54	43.12	38.80	63.29	46.86	42.42
DEVIANT	–	**24.63**	**16.54**	**14.52**	**32.60**	**23.04**	**19.99**	**61.00**	**46.00**	**40.18**	**65.28**	**49.63**	**43.50**

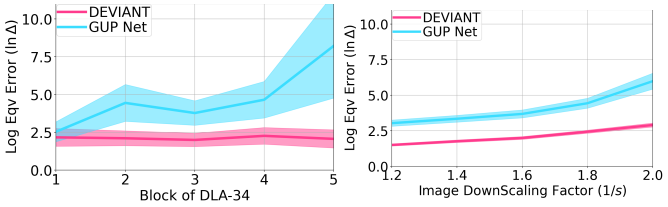
forms exceedingly well in the cross-dataset evaluation than [4, 49, 67]. We believe this happens because [4, 49, 67] rely on data or geometry to get the depth, while DEVIANT is equivariant to the depth translations, and therefore, outputs consistent depth. So, DEVIANT is more robust to data distribution changes.

Alternatives to Equivariance. We now compare with alternatives to equivariance in the following paragraphs.

(a) Scale Augmentation. A withstanding question in machine learning is the choice between equivariance and data augmentation [25]. Tab. 7 compares scale equivariance and scale augmentation. GUP Net [49] uses scale-augmentation and therefore, Tab. 7 shows that equivariance also benefits models which use scale-augmentation. This agrees with Tab. 2 of [74], where they observe that both augmentation and equivariance benefits classification on MNIST-scale dataset.

(b) Other Equivariant Architectures. We now benchmark adding depth (scale) equivariance to a 2D translation equivariant CNN and a transformer which learns the equivariance. Therefore, we compare DEVIANT with GUP Net [49] (a CNN), and DETR3D [85] (a transformer) in Tab. 8. As DETR3D does not report KITTI results, we trained DETR3D on KITTI using their public code. DEVIANT outperforms GUP Net and also surpasses DETR3D by a large margin. This happens because learning equivariance requires more data [90] compared to architectures which hardcode equivariance like CNN or DEVIANT.

(c) Dilated Convolution. DEVIANT adjusts the receptive field based on the object scale, and so, we compare with the dilated CNN (DCNN) [97] and D4LCN



(a) At blocks (depths) of backbone. (b) Varying scaling factors.

Fig. 4: **Log Equivariance Error** (Δ) comparison for DEVIANT and GUP Net at **(a)** different blocks with random image scaling factors **(b)** different image scaling factors at depth 3. DEVIANT shows **lower** scale equivariance error than vanilla GUP Net [49].

[20] in Tab. 9. The results show that DCNN performs sub-par to DEVIANT. This is expected because dilation corresponds to integer scales [92] while the scaling is generally a float in monocular detection. D4LCN [20] uses monocular depth as input to adjust the receptive field. DEVIANT (without depth) also outperforms D4LCN on Hard cars, which are more distant.

(d) Other Convolutions. We now compare with other known convolutions in literature such as Log-polar convolution [106], Dilated convolution [97] convolution and DISCO [72] in Tab. 11. The results show that the log-polar convolution does not work well, and SES convolutions are better suited to embed depth (scale) equivariance. As described in Sec. 4, we investigate the behavior of log-polar convolution through a small experiment. We calculate the SSIM [86] of the original image and the image obtained after the upscaling, log-polar, inverse log-polar, and downscaling blocks. We then average the SSIM over all KITTI Val images. We repeat this experiment for multiple image heights and scaling factors. The ideal SSIM should have been one. However, Fig. 2c shows that SSIM does not reach 1 even after upscaling by 4. This result confirms that log-polar convolution loses information at low resolutions resulting in inaccurate detection.

Next, the results show that dilated convolution [97] performs sub-par to DEVIANT. Moreover, DISCO [72] also does not outperform SES convolution which agrees with the 2D tracking results of [72].

(e) Feature Pyramid Network (FPN). Our baseline GUP Net [49] uses FPN [44] and Tab. 5 shows that DEVIANT outperforms GUP Net. Hence, we conclude that equivariance also benefits models which use FPN.

Comparison of Equivariance Error. We next quantitatively evaluate the scale equivariance of DEVIANT vs. GUP Net [49], using the equivariance error metric [74]. The equivariance error Δ is the normalized difference between the scaled feature map and the feature map of the scaled image, and is given by $\Delta = \frac{1}{N}\sum_{i=1}^{N} \frac{||\mathcal{T}_{s_i}\Phi(h_i) - \Phi(\mathcal{T}_{s_i}h_i)||_2^2}{||\mathcal{T}_{s_i}\Phi(h_i)||_2^2}$, where Φ denotes the neural network, $\mathcal{T}_{s_i}$ is the scaling transformation for the image i, and N is the total number of images. The equivariance error is zero if the scale equivariance is perfect. We plot the log of this error at different blocks of DEVIANT and GUP Net backbones and also plot at different downscaling of KITTI Val images in Fig. 4. The plots show that DEVIANT has low equivariance error than GUP Net. This is expected since the

Table 10: **3D and 2D detection** on KITTI Val cars.

Method	IoU ≥ 0.7						IoU ≥ 0.5					
	$AP_{3D\|R_{40}}$[%](↑)			$AP_{2D\|R_{40}}$[%](↑)			$AP_{3D\|R_{40}}$[%](↑)			$AP_{2D\|R_{40}}$[%](↑)		
	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
GUP Net [49]	21.10	15.48	12.88	96.78	88.87	79.02	58.95	43.99	38.07	99.52	91.89	81.99
DEVIANT (Ours)	24.63	16.54	14.52	96.68	88.66	78.87	61.00	46.00	40.18	97.12	91.77	81.93

Table 11: **Ablation studies** on KITTI Val cars.

Change from DEVIANT :		$IoU_{3D} \geq 0.7$						$IoU_{3D} \geq 0.5$					
Changed	From ⟶▶To	$AP_{3D\|R_{40}}$[%](↑)			$AP_{BEV\|R_{40}}$[%](↑)			$AP_{3D\|R_{40}}$[%](↑)			$AP_{BEV\|R_{40}}$[%](↑)		
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
Convolution	SES⟶▶Vanilla	21.10	15.48	12.88	28.58	20.92	17.83	58.95	43.99	38.07	64.60	47.76	42.97
	SES⟶▶Log-polar [106]	9.19	6.77	5.78	16.39	11.15	9.80	40.51	27.62	23.90	45.66	31.34	25.80
	SES⟶▶Dilated [97]	21.66	15.49	12.90	30.22	22.06	19.01	57.54	43.12	38.80	63.29	46.86	42.42
	SES⟶▶DISCO [72]	20.21	13.84	11.46	28.56	19.38	16.41	55.22	39.76	35.37	59.46	43.16	38.52
Downscale	10% ⟶▶ 5%	24.24	16.51	14.43	31.94	22.86	19.82	60.64	44.46	40.02	64.68	49.30	43.49
α	10% ⟶▶ 20%	22.19	15.85	13.48	31.15	23.01	19.90	61.24	44.93	40.22	67.46	50.10	43.83
BNP	SE⟶▶ Vanilla	24.39	16.20	14.36	32.43	22.53	19.70	62.81	46.14	40.38	67.87	50.23	44.08
Scales	3 ⟶▶ 1	23.20	16.29	13.63	31.76	23.23	19.97	61.90	46.66	40.61	67.37	50.31	43.93
	3 ⟶▶ 2	24.15	16.48	14.55	32.42	23.17	20.07	61.05	46.34	40.46	67.36	50.32	44.07
—	**DEVIANT (best)**	**24.63**	**16.54**	**14.52**	**32.60**	**23.04**	**19.99**	**61.00**	**46.00**	**40.18**	**65.28**	**49.63**	**43.50**

feature maps of the proposed DEVIANT are additionally equivariant to scale transformations (depth translations). We also visualize the equivariance error for a validation image and for the objects of this image in Fig. 12 in the supplementary. The qualitative plots also show a lower error for the proposed DEVIANT, which agrees with Fig. 4. Fig. 12a shows that equivariance error is particularly low for nearby cars which also justifies the good performance of DEVIANT on Easy (nearby) cars in Tabs. 3 and 5.

Does 2D Detection Suffer? We now investigate whether 2D detection suffers from using DEVIANT backbones in Tab. 10. The results show that DEVIANT introduces minimal decrease in the 2D detection performance. This is consistent with [73], who report that 2D tracking improves with the SE networks.

Ablation Studies. Tab. 11 compares the modifications of our approach on KITTI Val cars based on the experimental settings of Sec. 5.

(a) Floating or Integer Downscaling? We next investigate the question that whether one should use floating or integer downscaling factors for DEVIANT. We vary the downscaling factors as $(1{+}2\alpha, 1{+}\alpha, 1)$ and therefore, our scaling factor $s = \left(\frac{1}{1+2\alpha}, \frac{1}{1+\alpha}, 1\right)$. We find that α of 10% works the best. We again bring up the dilated convolution (Dilated) results at this point because dilation is a scale equivariant operation for integer downscaling factors [92] ($\alpha = 100\%, s = 0.5$). Tab. 11 results suggest that the downscaling factors should be floating numbers.

(b) SE BNP. As described in Sec. 4, we ablate DEVIANT against the case when only convolutions are SE but BNP layers are not. So, we place Scale-Projection [74] immediately after every SES convolution. Tab. 11 shows that such a network performs slightly sub-optimal to our final model.

(c) Number of Scales. We next ablate against the usage of Hermite scales. Using three scales performs better than using only one scale especially on Mod and Hard objects, and slightly better than using two scales.

Table 12: **Results on Waymo Val vehicles**. [Key: **Best**, **Second Best**]

IoU_{3D}	Difficulty	Method	Extra	AP_{3D} [%](↑)				APH_{3D} [%](↑)			
				All	0-30	30-50	50-∞	All	0-30	30-50	50-∞
0.7	Level_1	CaDDN [62]	LiDAR	5.03	14.54	1.47	0.10	4.99	14.43	1.45	0.10
		PatchNet [50] in [80]	Depth	0.39	1.67	0.13	0.03	0.39	1.63	0.12	0.03
		PCT [80]	Depth	0.89	3.18	0.27	0.07	0.88	3.15	0.27	0.07
		M3D-RPN [4] in [62]	–	0.35	1.12	0.18	0.02	0.34	1.10	0.18	0.02
		GUP Net (Retrained) [49]	–	**2.28**	**6.15**	**0.81**	**0.03**	**2.27**	**6.11**	**0.80**	**0.03**
		DEVIANT (Ours)	–	**2.69**	**6.95**	**0.99**	**0.02**	**2.67**	**6.90**	**0.98**	**0.02**
0.7	Level_2	CaDDN [62]	LiDAR	4.49	14.50	1.42	0.09	4.45	14.38	1.41	0.09
		PatchNet [50] in [80]	Depth	0.38	1.67	0.13	0.03	0.36	1.63	0.11	0.03
		PCT [80]	Depth	0.66	3.18	0.27	0.07	0.66	3.15	0.26	0.07
		M3D-RPN [4] in [62]	–	0.33	1.12	0.18	**0.02**	0.33	1.10	0.17	**0.02**
		GUP Net (Retrained) [49]	–	**2.14**	**6.13**	**0.78**	**0.02**	**2.12**	**6.08**	**0.77**	**0.02**
		DEVIANT (Ours)	–	**2.52**	**6.93**	**0.95**	**0.02**	**2.50**	**6.87**	**0.94**	**0.02**
0.5	Level_1	CaDDN [62]	LiDAR	17.54	45.00	9.24	0.64	17.31	44.46	9.11	0.62
		PatchNet [50] in [80]	Depth	2.92	10.03	1.09	0.23	2.74	9.75	0.96	0.18
		PCT [80]	Depth	4.20	14.70	1.78	0.39	4.15	14.54	1.75	0.39
		M3D-RPN [4] in [62]	–	3.79	11.14	2.16	**0.26**	3.63	10.70	2.09	**0.21**
		GUP Net (Retrained) [49]	–	**10.02**	**24.78**	**4.84**	**0.22**	**9.94**	**24.59**	**4.78**	**0.22**
		DEVIANT (Ours)	–	**10.98**	**26.85**	**5.13**	0.18	**10.89**	**26.64**	**5.08**	0.18
0.5	Level_2	CaDDN [62]	LiDAR	16.51	44.87	8.99	0.58	16.28	44.33	8.86	0.55
		PatchNet [50] in [80]	Depth	2.42	10.01	1.07	0.22	2.28	9.73	0.97	0.16
		PCT [80]	Depth	4.03	14.67	1.74	0.36	4.15	14.51	1.71	0.35
		M3D-RPN [4] in [62]	–	3.61	11.12	2.12	**0.24**	3.46	10.67	2.04	**0.20**
		GUP Net (Retrained) [49]	–	**9.39**	**24.69**	**4.67**	**0.19**	**9.31**	**24.50**	**4.62**	**0.19**
		DEVIANT (Ours)	–	**10.29**	**26.75**	**4.95**	0.16	**10.20**	**26.54**	**4.90**	0.16

5.3 Waymo Val Monocular 3D Detection

We also benchmark our method on the Waymo dataset [75] which has more variability than KITTI. Tab. 12 shows the results on Waymo Val split. The results show that DEVIANT outperforms the baseline GUP Net [49] on multiple levels and multiple thresholds. The biggest gains are on the nearby objects which is consistent with Tabs. 3 and 5. Interestingly, DEVIANT also outperforms PatchNet [50] and PCT [80] without using depth. Although the performance of DEVIANT lags CaDDN [62], it is important to stress that CaDDN uses LiDAR data in training, while DEVIANT is an image-only method.

6 Conclusions

This paper studies the modeling error in monocular 3D detection in detail and takes the first step towards convolutions equivariant to arbitrary 3D translations in the projective manifold. Since the depth is the hardest to estimate for this task, this paper proposes Depth EquiVarIAnt NeTwork (DEVIANT) built with existing scale equivariant steerable blocks. As a result, DEVIANT is equivariant to the depth translations in the projective manifold whereas vanilla networks are not. The additional depth equivariance forces the DEVIANT to learn consistent depth estimates and therefore, DEVIANT achieves SOTA detection results on KITTI and Waymo datasets in the image-only category and performs competitively to methods using extra information. Moreover, DEVIANT works better than vanilla networks in cross-dataset evaluation. Future works include applying the idea to Pseudo-LiDAR [83], and monocular 3D tracking.

References

1. The KITTI Vision Benchmark Suite. http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d, accessed: 2022-07-03 8
2. Alhaija, H., Mustikovela, S., Mescheder, L., Geiger, A., Rother, C.: Augmented reality meets computer vision: Efficient data generation for urban driving scenes. IJCV (2018) 1
3. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020) 2
4. Brazil, G., Liu, X.: M3D-RPN: Monocular 3D region proposal network for object detection. In: ICCV (2019) 1, 2, 4, 10, 11, 14, 27, 34, 35
5. Brazil, G., Pons-Moll, G., Liu, X., Schiele, B.: Kinematic 3D object detection in monocular video. In: ECCV (2020) 1, 4, 9, 10, 24, 27, 36
6. Bronstein, M.: Convolution from first principles. https://towardsdatascience.com/deriving-convolution-from-first-principles-4ff124888028, accessed: 2021-08-13 1, 3, 4, 21
7. Bronstein, M., Bruna, J., Cohen, T., Veličković, P.: Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. arXiv preprint arXiv:2104.13478 (2021) 3, 4, 21
8. Burns, B., Weiss, R., Riseman, E.: The non-existence of general-case view-invariants. In: Geometric invariance in computer vision (1992) 4, 5, 21, 22
9. Caesar, H., Bankiti, V., Lang, A., Vora, S., Liong, V., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: CVPR (2020) 8, 10, 34
10. Chabot, F., Chaouch, M., Rabarisoa, J., Teuliere, C., Chateau, T.: Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image. In: CVPR (2017) 4
11. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3D object detection for autonomous driving. In: CVPR (2016) 4
12. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A., Ma, H., Fidler, S., Urtasun, R.: 3D object proposals for accurate object class detection. In: NeurIPS (2015) 8
13. Chen, Y., Tai, L., Sun, K., Li, M.: MonoPair: Monocular 3D object detection using pairwise spatial relationships. In: CVPR (2020) 1, 4, 9
14. Chong, Z., Ma, X., Zhang, H., Yue, Y., Li, H., Wang, Z., Ouyang, W.: MonoDistill: Learning spatial features for monocular 3D object detection. In: ICLR (2022) 9, 10, 37
15. Cohen, T., Geiger, M., Köhler, J., Welling, M.: Spherical CNNs. In: ICLR (2018) 2, 3
16. Cohen, T., Welling, M.: Learning the irreducible representations of commutative lie groups. In: ICML (2014) 3
17. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: ICML (2016) 3, 21
18. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR (2009) 30
19. Dieleman, S., De Fauw, J., Kavukcuoglu, K.: Exploiting cyclic symmetry in convolutional neural networks. In: ICML (2016) 3, 21
20. Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., Luo, P.: Learning depth-guided convolutions for monocular 3D object detection. In: CVPR Workshops (2020) 4, 11, 12, 27

21. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 3
22. Esteves, C., Allen-Blanchette, C., Zhou, X., Daniilidis, K.: Polar transformer networks. In: ICLR (2018) 3, 4
23. Fidler, S., Dickinson, S., Urtasun, R.: 3D object detection and viewpoint estimation with a deformable 3D cuboid model. In: NeurIPS (2012) 4
24. Freeman, W., Adelson, E.: The design and use of steerable filters. TPAMI (1991) 4, 7
25. Gandikota, K., Geiping, J., Lähner, Z., Czapliński, A., Moeller, M.: Training or architecture? how to incorporate invariance in neural networks. arXiv preprint arXiv:2106.10044 (2021) 2, 11, 21
26. Ganea, O.E., Bécigneul, G., Hofmann, T.: Hyperbolic neural networks. In: NeurIPS (2017) 2, 3
27. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. IJRR (2013) 38
28. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR (2012) 8
29. Ghosh, R., Gupta, A.: Scale steerable filters for locally scale-invariant convolutional neural networks. In: ICML Workshops (2019) 3, 4, 6, 7, 29
30. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003) 3, 4, 5, 23, 39
31. Henriques, J., Vedaldi, A.: Warped convolutions: Efficient invariance to spatial transformations. In: ICML (2017) 3
32. Jansson, Y., Lindeberg, T.: Scale-invariant scale-channel networks: Deep networks that generalise to previously unseen scales. IJCV (2021) 3, 4, 6
33. Jing, L.: Physical symmetry enhanced neural networks. Ph.D. thesis, Massachusetts Institute of Technology (2020) 3
34. Kanazawa, A., Sharma, A., Jacobs, D.: Locally scale-invariant convolutional neural networks. In: NeurIPS Workshops (2014) 3
35. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 31
36. Kumar, A., Brazil, G., Liu, X.: GrooMeD-NMS: Grouped mathematically differentiable NMS for monocular 3D object detection. In: CVPR (2021) 4, 7, 9, 10, 32, 34, 37, 38
37. Kumar, A., Marks, T., Mou, W., Wang, Y., Jones, M., Cherian, A., Koike-Akino, T., Liu, X., Feng, C.: LUVLi face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In: CVPR (2020) 4
38. Kumar, A., Prabhakaran, V.: Estimation of bandlimited signals from the signs of noisy samples. In: ICASSP (2013) 7
39. Lambert, J., Liu, Z., Sener, O., Hays, J., Koltun, V.: MSeg: A composite dataset for multi-domain semantic segmentation. In: CVPR (2020) 36
40. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE (1998) 3, 4
41. Lee, J., Han, M., Ko, D., Suh, I.: From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326 (2019) 35, 36, 39
42. Li, P., Zhao, H., Liu, P., Cao, F.: RTM3D: Real-time monocular 3D detection from object keypoints for autonomous driving. In: ECCV (2020) 1, 4

43. Lian, Q., Ye, B., Xu, R., Yao, W., Zhang, T.: Geometry-aware data augmentation for monocular 3D object detection. arXiv preprint arXiv:2104.05858 (2021) 2, 4
44. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017) 12, 30
45. Liu, L., Lu, J., Xu, C., Tian, Q., Zhou, J.: Deep fitting degree scoring network for monocular 3D object detection. In: CVPR (2019) 1, 4
46. Liu, X., Xue, N., Wu, T.: Learning auxiliary monocular contexts helps monocular 3D object detection. In: AAAI (2022) 4
47. Liu, Y., Yixuan, Y., Liu, M.: Ground-aware monocular 3D object detection for autonomous driving. Robotics and Automation Letters (2021) 2, 9, 10
48. Liu, Z., Zhou, D., Lu, F., Fang, J., Zhang, L.: AutoShape: Real-time shape-aware monocular 3D object detection. In: ICCV (2021) 4, 9
49. Lu, Y., Ma, X., Yang, L., Zhang, T., Liu, Y., Chu, Q., Yan, J., Ouyang, W.: Geometry uncertainty projection network for monocular 3D object detection. In: ICCV (2021) 1, 2, 4, 7, 8, 9, 10, 11, 12, 13, 14, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
50. Ma, X., Liu, S., Xia, Z., Zhang, H., Zeng, X., Ouyang, W.: Rethinking Pseudo-LiDAR representation. In: ECCV (2020) 4, 14
51. Ma, X., Ouyang, W., Simonelli, A., Ricci, E.: 3D object detection from images for autonomous driving: A survey. arXiv preprint arXiv:2202.02980 (2022) 4
52. Ma, X., Wang, Z., Li, H., Zhang, P., Ouyang, W., Fan, X.: Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving. In: ICCV (2019) 4
53. Ma, X., Zhang, Y., Xu, D., Zhou, D., Yi, S., Li, H., Ouyang, W.: Delving into localization errors for monocular 3D object detection. In: CVPR (2021) 2, 8, 10, 25, 27, 28, 32
54. Marcos, D., Kellenberger, B., Lobry, S., Tuia, D.: Scale equivariance in CNNs with vector fields. In: ICML Workshops (2018) 3
55. Marcos, D., Volpi, M., Komodakis, N., Tuia, D.: Rotation equivariant vector field networks. In: ICCV (2017) 3
56. Micheli, A.: Neural network for graphs: A contextual constructive approach. IEEE Transactions on Neural Networks (2009) 3
57. Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is Pseudo-LiDAR needed for monocular 3D object detection? In: ICCV (2021) 4, 8, 9, 10, 33, 37
58. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019) 8
59. Payet, N., Todorovic, S.: From contours to 3D object detection and pose estimation. In: ICCV (2011) 4
60. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Multi-view and 3D deformable part models. TPAMI (2015) 4
61. Rath, M., Condurache, A.: Boosting deep neural networks with geometrical prior knowledge: A survey. arXiv preprint arXiv:2006.16867 (2020) 1, 3, 4, 21
62. Reading, C., Harakeh, A., Chae, J., Waslander, S.: Categorical depth distribution network for monocular 3D object detection. In: CVPR (2021) 4, 8, 9, 10, 14, 30, 32, 37
63. Rematas, K., Kemelmacher-Shlizerman, I., Curless, B., Seitz, S.: Soccer on your tabletop. In: CVPR (2018) 1

64. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NeurIPS (2015) 1
65. Saxena, A., Driemeyer, J., Ng, A.: Robotic grasping of novel objects using vision. IJRR (2008) 1
66. Shi, S., Wang, X., Li, H.: PointRCNN: 3D object proposal generation and detection from point cloud. In: CVPR (2019) 4
67. Shi, X., Ye, Q., Chen, X., Chen, C., Chen, Z., Kim, T.K.: Geometry-based distance decomposition for monocular 3D object detection. In: ICCV (2021) 2, 8, 9, 10, 11, 34, 35, 37, 39
68. Simonelli, A., Bulò, S., Porzi, L., Antequera, M., Kontschieder, P.: Disentangling monocular 3D object detection: From single to multi-class recognition. TPAMI (2020) 1, 8, 9, 34, 37
69. Simonelli, A., Bulò, S., Porzi, L., Kontschieder, P., Ricci, E.: Are we missing confidence in Pseudo-LiDAR methods for monocular 3D object detection? In: ICCV (2021) 4, 9, 10, 36
70. Simonelli, A., Bulò, S., Porzi, L., López-Antequera, M., Kontschieder, P.: Disentangling monocular 3D object detection. In: ICCV (2019) 8, 34
71. Simonelli, A., Bulò, S., Porzi, L., Ricci, E., Kontschieder, P.: Towards generalization across depth for monocular 3D object detection. In: ECCV (2020) 1, 2, 4
72. Sosnovik, I., Moskalev, A., Smeulders, A.: DISCO: accurate discrete scale convolutions. In: BMVC (2021) 12, 13
73. Sosnovik, I., Moskalev, A., Smeulders, A.: Scale equivariance improves siamese tracking. In: WACV (2021) 4, 6, 8, 13, 29, 30
74. Sosnovik, I., Szmaja, M., Smeulders, A.: Scale-equivariant steerable networks. In: ICLR (2020) 3, 4, 6, 7, 11, 12, 13, 26, 29, 37
75. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020) 8, 14
76. Tang, Y., Dorn, S., Savani, C.: Center3D: Center-based monocular 3D object detection with joint depth understanding. arXiv preprint arXiv:2005.13423 (2020) 1, 2, 4
77. Thayalan-Vaz, S., M, S., Santhakumar, K., Ravi Kiran, B., Gauthier, T., Yogamani, S.: Exploring 2D data augmentation for 3D monocular object detection. arXiv preprint arXiv:2104.10786 (2021) 2, 4
78. Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., Riley, P.: Tensor field networks: Rotation-and translation-equivariant neural networks for 3D point clouds. arXiv preprint arXiv:1802.08219 (2018) 3
79. Wang, L., Du, L., Ye, X., Fu, Y., Guo, G., Xue, X., Feng, J., Zhang, L.: Depth-conditioned dynamic message propagation for monocular 3D object detection. In: CVPR (2021) 9, 10
80. Wang, L., Zhang, L., Zhu, Y., Zhang, Z., He, T., Li, M., Xue, X.: Progressive coordinate transforms for monocular 3D object detection. In: NeurIPS (2021) 8, 9, 10, 14, 37, 39
81. Wang, R., Walters, R., Yu, R.: Incorporating symmetry into deep dynamics models for improved generalization. In: ICLR (2021) 3
82. Wang, X., Zhang, S., Yu, Z., Feng, L., Zhang, W.: Scale-equalizing pyramid convolution for object detection. In: CVPR (2020) 33

83. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.: Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving. In: CVPR (2019) 4, 14
84. Wang, Y., Chen, X., You, Y., Li, L., Hariharan, B., Campbell, M., Weinberger, K., Chao, W.L.: Train in Germany, test in the USA: Making 3D object detectors generalize. In: CVPR (2020) 35
85. Wang, Y., Guizilini, V., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: DETR3D: 3D object detection from multi-view images via 3D-to-2D queries. In: CoRL (2021) 11, 39
86. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. TIP (2004) 12
87. Weiler, M., Forré, P., Verlinde, E., Welling, M.: Coordinate independent convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds. arXiv preprint arXiv:2106.06020 (2021) 3
88. Weiler, M., Hamprecht, F., Storath, M.: Learning steerable filters for rotation equivariant CNNs. In: CVPR (2018) 3
89. Wilk, M.v.d., Bauer, M., John, S., Hensman, J.: Learning invariances using the marginal likelihood. In: NeurIPS (2018) 3
90. Worrall, D., Brostow, G.: Cubenet: Equivariance to 3D rotation and translation. In: ECCV (2018) 3, 4, 11
91. Worrall, D., Garbin, S., Turmukhambetov, D., Brostow, G.: Harmonic networks: Deep translation and rotation equivariance. In: CVPR (2017) 3
92. Worrall, D., Welling, M.: Deep scale-spaces: Equivariance over scale. In: NeurIPS (2019) 4, 12, 13, 27
93. Wu, Y., Johnson, J.: Rethinking "batch" in batchnorm. arXiv preprint arXiv:2105.07576 (2021) 33
94. Xu, Y., Xiao, T., Zhang, J., Yang, K., Zhang, Z.: Scale-invariant convolutional neural networks. arXiv preprint arXiv:1411.6369 (2014) 4
95. Yang, G., Ramanan, D.: Upgrading optical flow to 3D scene flow through optical expansion. In: CVPR (2020) 8
96. Yeh, R., Hu, Y.T., Schwing, A.: Chirality nets for human pose regression. NeurIPS (2019) 3
97. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (2015) 11, 12, 13, 27
98. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: CVPR (2018) 30
99. Zhang, Y., Ma, X., Yi, S., Hou, J., Wang, Z., Ouyang, W., Xu, D.: Learning geometry-guided depth via projective modeling for monocular 3D object detection. arXiv preprint arXiv:2107.13931 (2021) 2
100. Zhang, Y., Lu, J., Zhou, J.: Objects are different: Flexible monocular 3D object detection. In: CVPR (2021) 1, 4, 9, 10, 37
101. Zhou, A., Knowles, T., Finn, C.: Meta-learning symmetries by reparameterization. In: ICLR (2021) 3
102. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019) 1
103. Zhou, Y., He, Y., Zhu, H., Wang, C., Li, H., Jiang, Q.: MonoEF: Extrinsic parameter free monocular 3D object detection. TPAMI (2021) 2, 4, 9, 10, 37
104. Zhu, W., Qiu, Q., Calderbank, R., Sapiro, G., Cheng, X.: Scale-equivariant neural networks with decomposed convolutional filters. arXiv preprint arXiv:1909.11193 (2019) 3, 4, 6, 7, 37

105. Zou, Z., Ye, X., Du, L., Cheng, X., Tan, X., Zhang, L., Feng, J., Xue, X., Ding, E.: The devil is in the task: Exploiting reciprocal appearance-localization features for monocular 3D object detection. In: ICCV (2021) 9
106. Zwicke, P., Kiss, I.: A new implementation of the mellin transform and its application to radar classification of ships. TPAMI (1983) 2, 3, 7, 12, 13