Geometry Uncertainty Projection Network for Monocular 3D Object Detection

Yan Lu[1,†,*] Xinzhu Ma[1,*] Lei Yang[2] Tianzhu Zhang[3]
Yating Liu[4] Qi Chu[3,✉] Junjie Yan[2] Wanli Ouyang[1,✉]
[1]The University of Sydney, SenseTime Computer Vision Group [2]Sensetime Group Limited
[3]School of Information Science and Technology, University of Science and Technology of China
[4]School of Data Science, University of Science and Technology of China
{yan.lu1, xinzhu.ma, wanli.ouyang}@sydney.edu.au {yanglei, yanjunjie}@sensetime.com
{tzzhang, qchu}@ustc.edu.cn liuyat@mail.ustc.edu.cn

Abstract

Geometry Projection is a powerful depth estimation method in monocular 3D object detection. It estimates depth dependent on heights, which introduces mathematical priors into the deep model. But projection process also introduces the error amplification problem, in which the error of the estimated height will be amplified and reflected greatly at the output depth. This property leads to uncontrollable depth inferences and also damages the training efficiency. In this paper, we propose a Geometry Uncertainty Projection Network (GUP Net) to tackle the error amplification problem at both inference and training stages. Specifically, a GUP module is proposed to obtains the geometry-guided uncertainty of the inferred depth, which not only provides high reliable confidence for each depth but also benefits depth learning. Furthermore, at the training stage, we propose a Hierarchical Task Learning strategy to reduce the instability caused by error amplification. This learning algorithm monitors the learning situation of each task by a proposed indicator and adaptively assigns the proper loss weights for different tasks according to their pre-tasks situation. Based on that, each task starts learning only when its pre-tasks are learned well, which can significantly improve the stability and efficiency of the training process. Extensive experiments demonstrate the effectiveness of the proposed method. The overall model can infer more reliable object depth than existing methods and outperforms the state-of-the-art image-based monocular 3D detectors by 3.74% and 4.7% AP_{40} of the car and pedestrian categories on the KITTI benchmark. The code and model will be released at https://github.com/SuperMHP/GUPNet.

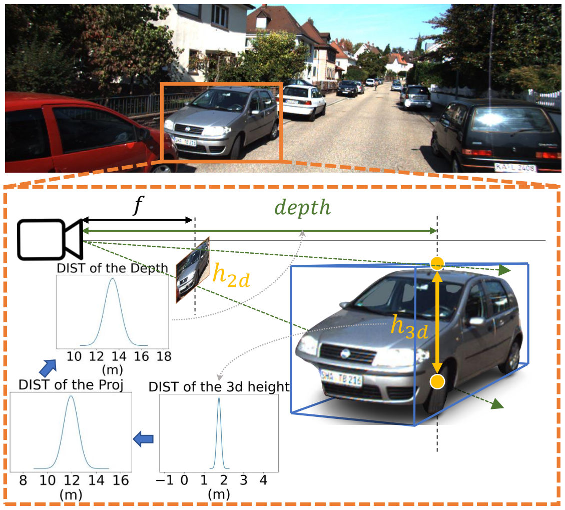

Figure 1. The main pipeline of our Geometry Uncertainty Projection module. The projection process is modeled in the probability framework. The inference depths can be represented as a distribution so that can provide both accurate values and scores.

1. Introduction

3D object detection is an important component in autonomous driving and has received increasing attention in recent years. Compared with the LiDAR/stereo-based methods [32, 35, 37, 40, 41, 49, 57], monocular 3D object detection is still a challenging task due to the lack of depth cues, which makes monocular object-level depth estimation naturally ill-posed. Therefore, the monocular 3D detector cannot achieve satisfactory performance even some complex network structures [39] are applied. Recently, to alleviate this problem, some works [36, 47] attempt to introduce geometry priors to help depth inference, of which a widely used prior is the perspective projection model.

Existing methods with the projection model usually es-

[†]This work was done when Yan Lu was an intern at SenseTime.
[*]Equal contribution.
[✉]Corresponding authors.

timate the height of 2D and 3D bounding box first and then infer the depth via the projection formula $depth = h_{3d} \cdot f / h_{2d}$ (f is the camera focal length). Depth inferred by this formula is highly related to the estimated 2D/3D heights so the error of the height estimation will also be reflected at the estimated depth. However, the error of height estimation is inevitable especially for the ill-posed 3D height estimation (2D height estimation is relatively more accurate because of the well-developed 2d detection), so we are more concerned about the depth inference error caused by the 3D height estimation error. To show the influence of this property, we visualize the depth shifts caused by a fixed 3D height error in Figure 2. We can find that a slight bias (0.1m) of 3D heights could cause a significant shift (even 4m) in the projected depth. This error amplification effect makes outputs of the projection-based methods hardly controllable, significantly affecting both inference reliability and training efficiency. In this paper, we propose a Geometry Uncertainty Projection Network that includes a Geometry Uncertainty Projection (GUP) module and a Hierarchical Task Learning (HTL) strategy to treat these problems.

The first problem is inference reliability. A small quality change in the 3D height estimation would cause a large change in the depth estimation quality. This makes the model cannot predict reliable uncertainty or confidence easily, leading to uncontrollable outputs. To tackle this problem, the GUP module is proposed to infer the depth based on the distribution form rather than a discrete value (see Figure 1). The depth distribution is inferred by the estimated 3D height distribution. So, the statistical characteristics of the estimated 3D height estimation would be reflected in the output depth distribution, which leads to more accurate Uncertainty. At the inference, this well-learned uncertainty would be mapped to a confidence value to indicate the depth inference quality, which makes the total projection process more reliable.

Another problem is the instability of model training. In particular, at the beginning of the training phase, the estimation of 2D/3D height tends to be noisy, and the errors will be amplified and cause outrageous depth estimation. Consequently, the training process of the network will be misled, which will lead to the degradation of the final performance. To solve the instability of the training, we propose the Hierarchical Task Learning (HTL) strategy, aiming to ensure that each task is trained only when all pre-tasks (*e.g.* 3D height estimation is one of the pre-tasks of depth estimation) are trained well. To achieve that, the HTL first measures the learning situation of each task by a well-designed learning situation indicator. Then it adjusts weights for each loss term automatically by the learning situation of their pre-tasks, which can significantly improve the training stability, thereby boosting the final performance.

In summary, the key contributions are as follows:

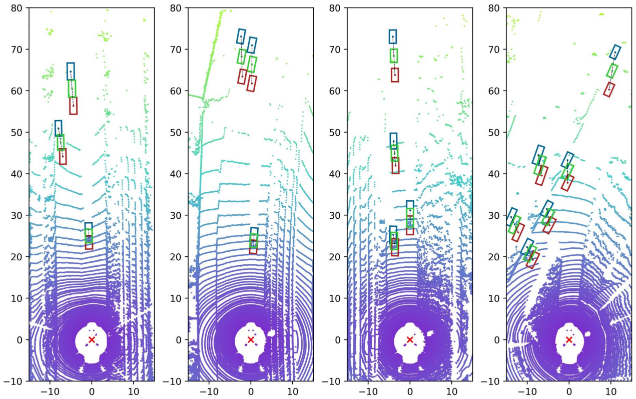

Figure 2. Visualized examples of the depth shift caused by ±0.1m 3D height jitter. We draw some bird's view examples to show the error amplification effect. In this figure, the unit of the horizontal axis and the vertical axis are both meters, and the vertical axis corresponds to the depth direction. The green boxes mean the original projection outputs. The blue and red boxes are shifted boxes caused by +0.1m and -0.1m 3D height bias respectively (best viewed in color).

- We propose a Geometry Uncertainty Projection (GUP) module combining both mathematical priors and uncertainty modeling, which significantly reduces the uncontrollable effect caused by the error amplification at the inference.
- For the training instability caused by task dependency in geometry-based methods, we propose a Hierarchical Task Learning (HTL) strategy, which can significantly improve the training efficiency.
- Evaluation on the challenging KITTI dataset shows the overall proposed GUP Net achieves state-of-the-art performance around 20.11% and 14.72% on the car and the pedestrian 3D detection respectively on the KITTI testing set.

2. Related works

Monocular 3D object detection. The monocular 3D object detection aims to predict 3D bounding boxes from a single given image [13, 17, 21, 25, 31, 43]. Existing methods focus on deep representation learning [39] and geometry priors [29, 30, 46, 56]. Deep3DBox [34] firstly tried to solve the key angle prediction problem by geometry priors. DeepMANTA [7] introduced the 3D CAD model to learn shape-based knowledge and guided to better dimension prediction results. GS3D [22] utilized the ROI surface features to extract better object representations. M3DRPN [4] gave a novel modified 3D anchor setting and proposed a depth-wise convolution to treat the monocular 3D detection task. MonoPair [10] proposed a pair-wise relationship to improve the monocular 3D detection performance.

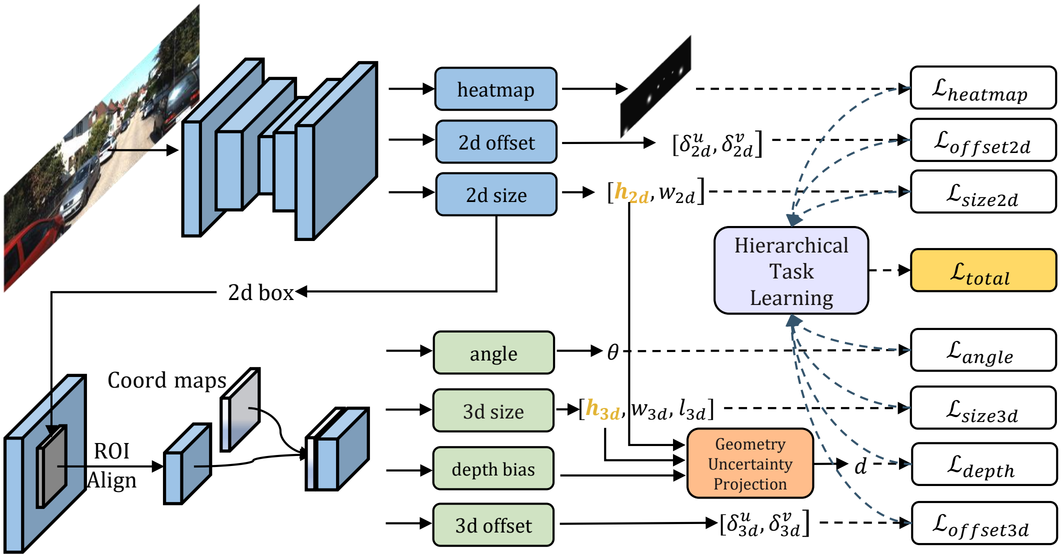

Figure 3. The framework of the Geometry Uncertainty Projection Network. The input image is sent to the network to extract the 2D box and basic 3D box parameters. And the Geometry Uncertainty Projection module would infer the depth based on the height parameters. And at the training, all task losses would be gathered by the Hierarchical Task Learning strategy to assign proper weights for them.

Except for these methods, many methods tried to introduce geometry projection to infer depth [1, 2, 6, 20]. Ivan *et al.* [2] combined the keypoint method and the projection to do geometry reasoning. Decoupled3D [6] used lengths of bounding box edges to project and get the inferred depth. Bao *et al.* [1] combined the center voting with the geometry projection to achieve better 3D center reasoning. All of these projection-based methods did not consider the error amplification problem, leading to the limited performance.

Uncertainty based depth estimation. The uncertainty theory is widely used in the deep regression method [3], which can model both aleatoric and epistemic uncertainty. This technology is well developed in the depth estimation [18, 24], which can significantly reduce the noise of the depth targets. However, these methods directly regressed the depth uncertainty by the deep models and neglected the relationships between the height and the depth. In this work, we try to compute the uncertainty via combining both end-to-end learning and the geometry relationships.

Multi-task learning. Multi-task learning is a widely studied topic in computer vision. Many works focus on task relation representation learning [28, 45, 48, 51, 52, 53]. Except that, some works also tried to adjust weights for different loss functions to solve the multi-task problem [11, 19, 54]. GradNorm [11] tried to solve the loss unbalance problem in joint multi-task learning and improved the training stability. Kendall *et al.* [19] proposed a task-uncertainty strategy to treat the task balance problems, which also achieved good results. These loss weights control methods assumed that each task is independent from each other, which are unsuitable for our method, since the multiple tasks in our framework form a hierarchical structure, i.e., some tasks are dependent on their pre-tasks. Therefore, we propose a Hierarchical Task Learning strategy to handle it.

3. Geometry Uncertainty Projection Network

Figure 3 shows the framework of the proposed Geometry Uncertainty Projection Network (GUP Net). Images are processed by the 2D detection backbone first, yielding 2D bounding boxes (Region of Interest, RoI) and then computes basic 3D bounding box information,*i.e.*, angle, dimensions and 3D projected center for each box. After that, the GUP module predicts the depth distribution via combining both mathematical priors and uncertainty modeling. This depth distribution provides an accurate inferred depth value and its corresponding uncertainty. The predicted uncertainty would be mapped to 3D detection confidence at the inference stage. Furthermore, to avoid misleading caused by the error amplification at the beginning of training, an efficient Hierarchical Task Learning (HTL) strategy would control the overall training process, where each task does not start training until its pre-tasks have been trained well.

3.1. 2D detection

As shown in Figure 3, our 2D detector is built on CenterNet [55], computing a heatmap to indicate the coarse locations and confidences of the objects and also predicting the 2D offset and 2D size for each potential 2D box.

3.2. RoI feature representation

To guide the model to focus on the object, we crop and resize the RoI features using RoIAlign [16]. However, those features lack location and size cues which are essential to the monocular depth estimation [12]. Therefore, we compute the normalized coordinate map, and then concatenate it with the feature maps of each RoI in a channel-wise manner to compensate for that cues (shown as Figure 3).

3.3. Basic 3D detection Heads

With the extracted RoI features, we construct several sub-heads on top of those features to predict some basic 3D bounding box information. A 3D offset branch aims to estimate the 3D center projection on the 2D feature maps [10]. The angle prediction branch predicts the relative alpha rotation angle [34]. And the 3D size branch estimates the 3D dimension parameters, including height, width and length. These predictions are supervised by $\mathcal{L}_{offset3d}$, $\mathcal{L}_{angle}$ and $\mathcal{L}_{size3d}$, respectively. Note that $\mathcal{L}_{size3d}$ includes three parts for different dimensions, *e.g.*, the height loss $\mathcal{L}_{h3d}$.

3.4. Geometry Uncertainty Projection

The basic 3D detection heads provide most information of the 3D bounding box except depth. Given the difficulty to regress depth directly, we propose a novel Geometry Uncertainty Projection model. The overall module builds the projection process in the probability framework rather than single values so that the model can compute better uncertainty for the inferred depth, which can indicate the depth inference reliability and also be helpful for model learning.

To achieve this goal, we first assume the prediction of the 3D height for each object is a Laplace distribution $La(\mu_h, \lambda_h)$[1]. The mean μ_h and the standard deviation σ_h are predicted by the 3D size stream in an end-to-end way. The μ_h denotes the regression target output and the σ_h is the uncertainty of the inference. Consequently, the 3D height loss function can be defined as:

$$\mathcal{L}_{h3d} = \frac{\sqrt{2}}{\sigma_h}|\mu_h - h_{3d}^{gt}| + log(\sigma_h). \tag{1}$$

The minimization of $\mathcal{L}_{h3d}$ make μ_h and the ground-truth height h_{3d}^{gt} as close as possible. Particularly, the difficult or noise-labeled samples usually incur large σ_{3d}, indicating the low prediction confidence. Based on the learned h_{3d} distribution, the depth distribution of the projection output $La(\mu_p, \lambda_p)$ can be approximated as:

$$\begin{aligned} d_p &= \frac{f \cdot h_{3d}}{h_{2d}} = \frac{f \cdot (\lambda_h \cdot X + \mu_h)}{h_{2d}} \\ &= \frac{f \cdot \lambda_h}{h_{2d}} \cdot X + \frac{f \cdot \mu_h}{h_{2d}}, \end{aligned} \tag{2}$$

where X is the standard Laplace distribution $La(0, 1)$. In this sense, the μ_p, σ_p are $\frac{f \cdot \mu_h}{h_{2d}}$ and $\frac{f \cdot \sigma_h}{h_{2d}}$, respectively. To obtain better predicted depth, we add a learned bias to modify the initial projection results. We also assume that the learned bias is a Laplace distribution $La(\mu_b, \sigma_b)$ and independent with the projection one. Accordingly, the final depth distribution can be written as:

$$\begin{aligned} d &= La(\mu_p, \lambda_p) + La(\mu_b, \lambda_b), \\ \mu_d &= \mu_p + \mu_b, \quad \sigma_d = \sqrt{(\sigma_p)^2 + (\sigma_b)^2}. \end{aligned} \tag{3}$$

We refer to the final uncertainty σ_d as Geometry based Uncertainty (GeU). This uncertainty reflects both the projection uncertainty and the bias learning uncertainty. With this formula, a small uncertainty of h_{3d} will be reflected in the GeU value. To optimize the final depth distribution, we apply the uncertainty regression loss:

$$\mathcal{L}_{depth} = \frac{\sqrt{2}}{\sigma_d}|\mu_d - d^{gt}| + log(\sigma_d). \tag{4}$$

Note that we also assume the depth distribution belong to Laplace distribution here for simplification. The overall loss would push the projection results close to the ground truth d^{gt} and the gradient would affect the depth bias, the 2D height and the 3D height simultaneously. Besides, the uncertainty of 3D height and depth bias is also trained in the optimization process.

During inference, the reliability of depth prediction is critical for real-world applications. A reliable inference system is expected to feedback high confidence for a good estimation and low score for a bad one. As our well-designed GeU has capability of indicating the uncertainty of depth, we further map its value to 0∼1 by an exponential function to indicate the depth Uncertainty-Confidence (UnC):

$$p_{depth} = exp(-\sigma_d). \tag{5}$$

It can provide more accurate confidence for each projection depth. Thus we use this confidence as the conditional 3D bounding box scores $p_{3d|2d}$ in the testing. The final inference score can be computed as:

$$p_{3d} = p_{3d|2d} \cdot p_{2d} = p_{depth} \cdot p_{2d}. \tag{6}$$

This score represents both the 2D detection confidence and the depth confidence, which can guide better reliability.

3.5. Hierarchical Task Learning

The GUP module mainly addresses the error amplification effect in the inference stage. Yet, this effect also damages the training procedure. Specifically, at the beginning of the training, the prediction of both h_{2d} and h_{3d} are far from accurate, which will mislead the overall training and damage the performance. To tackle this problem, we design

[1] The probability density function of a Laplace random variable $X \sim La(\mu, \lambda)$ is: $f_X(x) = \frac{1}{2\lambda}\exp(\frac{|x-\mu|}{\lambda})$, where μ and λ are Laplace parameters. The standard deviation σ can be computed by: $\sigma = \sqrt{2}\lambda$.

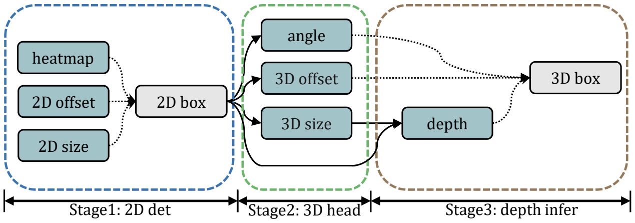

Figure 4. The task hierarchy of the GUP Net. The first stage is 2D detection. Built on top of RoI features, the second stage consists of basic 3D detection heads. Based on 2D and 3D heights estimated in the previous stages, the third stage infers the depth and then constitutes the 3D bounding box.

a Hierarchical Task Learning (HTL) to control weights for each task at each epoch. The overall loss is:

$$\mathcal{L}_{total} = \sum_{i \in \mathcal{T}} w_i(t) \cdot \mathcal{L}_i, \tag{7}$$

where $\mathcal{T}$ is the task set. t denotes the current epoch index and $\mathcal{L}_i$ means the i-th task loss function. $w_i(t)$ is the loss weight for the i-th task at the t-th epoch.

HTL is inspired by the motivation that each task should start training after its pre-task has been trained well. We split tasks into different stages as shown in Figure 4 and the loss weight $w_i(t)$ should be associated with all pre-tasks of the i-th task. The first stage is 2D detection, including heatmap, 2D offset, 2D size. Then, the second stage is the 3D heads containing angle, 3D offset and 3D size. All of these 3D tasks are built on the ROI features, so the tasks in 2D detection stage are their pre-tasks. Similarly, the final stage is the depth inference and its pre-tasks are the 3D size and all the tasks in 2D detection stage since depth prediction depends on the 3D height and 2D height. To train each task sufficiently, we aim to gradually increase the $w_i(t)$ from 0 to 1 as the training progresses. So we adopt the widely used polynomial time scheduling function [33] in the curriculum learning topic as our weighted function, which is adapted as follows:

$$w_i(t) = (\frac{t}{T})^{1-\alpha_i(t)}, \ \ \alpha_i(t) \in [0, 1], \tag{8}$$

where T is the total training epochs and the normalized time variable $\frac{t}{T}$ can automatically adjust the time scale. $\alpha_i(t)$ is an adjust parameter at the t-th epoch, corresponding to every pre-task of the i-th task. Figure 5 shows that α_i can change the trend of the time scheduler. The larger α_i is, the faster $w_i(\cdot)$ increases. From the definition of the adjust parameter, it is natural to decide its value via the learning situation of every pre-task. If all pre-task have been well trained, the α_i is expected be large, otherwise it should be small. This is motivated by the observation that human usually learn advanced courses after finishing fundamental courses. Therefore, $\alpha_i(t)$ is defined as:

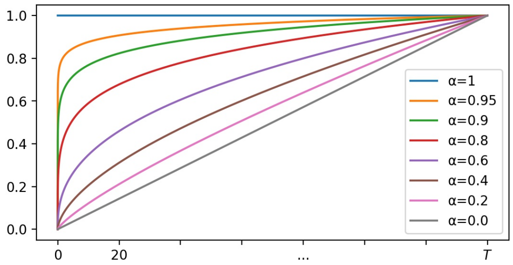

Figure 5. The polynomial time scheduling function with the adjust parameter. The vertical axis is the value of $w_i(t)$ and the horizontal axis is the epoch index t. (best viewed in color.)

$$\alpha_i(t) = \prod_{j \in \mathcal{P}_i} ls_j(t), \tag{9}$$

where $\mathcal{P}_i$ is the pre-task set for the i-th task. ls_j means the learning situation indicator of j-th task, which is a value between 0∼1. This formula means that α_i would get high values only when all pre-tasks have achieved high ls (trained well). For the ls_j, inspired by [11, 54], we design a scale-invariant factor to indicate the learning situation:

$$\begin{aligned} ls_j(t) &= \frac{\mathcal{DF}_j(K) - \mathcal{DF}_j(t)}{\mathcal{DF}_j(K)}, \\ \mathcal{DF}_j(t) &= \frac{1}{K} \sum_{\hat{t}=t-K}^{t-1} |\mathcal{L}'_j(\hat{t})|, \end{aligned} \tag{10}$$

where $\mathcal{L}'_j(\hat{t})$ is the derivative of the $\mathcal{L}_j(\cdot)$ at the $\hat{t}$-th epoch, which can indicate the local change trend of the loss function. The $\mathcal{DF}_j(t)$ computes the mean of derivatives in the recent K epochs before the t-th epoch to reflect the mean change trend. If the $\mathcal{L}_j$ drops quickly in the recent K epochs, the $\mathcal{DF}_j$ will get a large value. So the ls_j formula means comparing the difference between the current trend $\mathcal{DF}_j(t)$ and the trend of the first K epochs at the beginning of training $\mathcal{DF}_j(K)$ for the j-th task. If the current loss trend is similar to the beginning trend, the indicator will give a small value, which means that this task has not trained well. Conversely, if a task tends to converge, the ls_j will be close to 1, meaning that the learning situation of this task is satisfied.

Based on the overall design, the loss weight of each term can reflect the learning situation of its pre-tasks dynamically, which can make the training more stable.

4. Experiments

4.1. Setup

Dataset. The KITTI 3D dataset [15] is the most commonly used benchmark in the 3D object detection task, and it provides left camera images, calibration files, annotations for

Table 1. **3D object detection on the KITTI *test* set.** We highlight the best results in **bold**. For the extra data: 1). 'Depth' means the methods use extra depth annotations or off-the-shelf networks pre-trained from a larger depth estimation dataset. 2). 'Temporal' means using additional temporal data. 3). 'LiDAR' means utilizing real LiDAR data for better training. 4). 'None' denotes no extra data is used.

Method	Extra data	Car@IoU=0.7			Pedestrian@IoU=0.5			Cyclist@IoU=0.5		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
Mono-PLiDAR [47]	Depth	10.76	7.50	6.10	–	–	–	–	–	–
Decoupled-3D [6]	Depth	11.08	7.02	5.63	–	–	–	–	–	–
AM3D [30]	Depth	16.50	10.74	9.52	–	–	–	–	–	–
PatchNet [29]	Depth	15.68	11.12	10.17	–	–	–	–	–	–
DA-3Ddet [14]	Depth	16.77	11.50	8.93	–	–	–	–	–	–
D4LCN [13]	Depth	16.65	11.72	9.51	4.55	3.42	2.83	2.45	1.67	1.36
Kinematic [5]	Temporal	19.07	12.72	9.17	–	–	–	–	–	–
MonoPSR [20]	LiDAR	10.76	7.25	5.85	6.12	4.00	3.30	**8.70**	**4.74**	**3.68**
CaDNN [38]	LiDAR	19.17	13.41	11.46	12.87	8.14	6.76	7.00	3.41	3.30
MonoDIS [43]	None	10.37	7.94	6.40	–	–	–	–	–	–
UR3D [42]	None	15.58	8.61	6.00	–	–	–	–	–	–
M3D-RPN [4]	None	14.76	9.71	7.42	4.92	3.48	2.94	0.94	0.65	0.47
SMOKE [27]	None	14.03	9.76	7.84	–	–	–	–	–	–
MonoPair [10]	None	13.04	9.99	8.65	10.02	6.68	5.53	3.79	2.12	1.83
RTM3D [23]	None	14.41	10.34	8.77	–	–	–	–	–	–
MoVi-3D [44]	None	15.19	10.90	9.26	8.99	5.44	4.57	1.08	0.63	0.70
RAR-Net [26]	None	16.37	11.01	9.52	–	–	–	–	–	–
GUP Net (Ours)	None	**20.11**	**14.20**	**11.77**	**14.72**	**9.53**	**7.87**	4.18	2.65	2.09
Improvement	*vs.* Depth	+3.46	+2.48	+2.26	+10.17	+6.11	+5.04	+1.73	+0.98	+0.73
Improvement	*vs.* Temporal	+1.04	+1.48	+2.60	–	–	–	–	–	–
Improvement	*vs.* LiDAR	+0.94	+0.79	+0.31	+1.85	+1.39	+1.11	-4.52	-2.09	-2.09
Improvement	*vs.* None	+3.74	+3.19	+2.25	+4.7	+2.85	+2.34	+0.39	+0.53	+0.26

standard monocular 3D detection. It totally provides 7,481 frames for training and 7,518 frames for testing. Following [8, 9], we split the training data into a training set (3,712 images) and a validation set (3,769 images). We conduct ablation studies based on this split and also report the final results with the model trained on all 7,481 images and tested by KITTI official server.

Evaluation protocols. All the experiments follow the standard evaluation protocol in the monocular 3D object detection and bird's view (BEV) detection tasks. Following [43], we evaluate the AP_{40} to avoid the bias of original AP_{11}.

Implementation details. We use DLA-34 [50] as our backbone for both baseline and our method. The resolution of the input image is set to 380 × 1280 and the feature maps down-sampling rate is 4. Each 2D sub-head has two Conv layers (the channel of the first one is set to 256) and each 3D sub-head includes one 3x3 Conv layer with 256 channels, one averaged pooling layer and one fully-connected layer. The output channels of these heads are depending on the output data structure. We train our model with the batch-size of 32 on 3 Nvidia TiTan XP GPUs for 140 epochs. The initial learning rate is $1.25e^{-3}$, which is decayed by 0.1 at the 90-th and the 120-th epoch. To make the training more stable, we apply the linear warm-up strategy in the first 5 epochs. The K in the HTL is also set to 5.

4.2. Main Results

Results of Car category on the KITTI *test* set. As shown in Table 1, we first compare our method with other counterparts on the KITTI test set. Overall, the proposed method achieves superior results of Car category over previous methods, including those with extra data. Under fair conditions, our method achieves 3.74%, 3.19%, and 2.25% gains on the easy, moderate, and hard settings, respectively. Furthermore, our method also outperforms the methods with extra data. For instance, compared with the recently proposed CaDNN [38] utilizing LiDAR signals as supervision of depth estimation sub-task, our method still obtains 0.94%, 0.79%, and 0.31% gains on the three difficulty settings, which confirms the effectiveness of the proposed method.

Results of Car category on the KITTI *validation* set. We also present our model's performance on the KITTI validation set in Table 2 for better comparison, including different tasks and IoU thresholds. Specifically, our method gets almost the same performance as the best competing method MonoPair at the 0.5 IoU threshold. Moreover, our method improves with respect to MonoPair by 4.16%/4.77% for 3D/BEV detection under the moderate setting at 0.7 IoU threshold. This shows that our method is very suitable for high-precision tasks, which is a vital feature in the automatic driving scene. Note that RTM3D and RAR-Net do not report the AP_{40} metric on the validation set, and the comparison with them on AP_{11} metric can be found in supplementary materials.

Pedestrian/Cyclist detection on the KITTI *test* set. We also report the Pedestrian/Cyclist detection results in Table 1. Specifically, our method remarkably outperforms all

Table 2. **Performance of the Car category on the KITTI *validation* set.** We highlight the best results in **bold**.

Method	3D@IoU=0.7			BEV@IoU=0.7			3D@IoU=0.5			BEV@IoU=0.5		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
CenterNet [55]	0.60	0.66	0.77	3.46	3.31	3.21	20.00	17.50	15.57	34.36	27.91	24.65
MonoGRNet [36]	11.90	7.56	5.76	19.72	12.81	10.15	47.59	32.28	25.50	48.53	35.94	28.59
MonoDIS [43]	11.06	7.60	6.37	18.45	12.58	10.66	-	-		-	-	
M3D-RPN [4]	14.53	11.07	8.65	20.85	15.62	11.88	48.53	35.94	28.59	53.35	39.60	31.76
MoVi-3D [44]	14.28	11.13	9.68	22.36	17.87	15.73	-	-	-	-	-	-
MonoPair [10]	16.28	12.30	10.42	24.12	18.17	15.76	55.38	**42.39**	**37.99**	61.06	**47.63**	**41.92**
GUP Net (Ours)	**22.76**	**16.46**	**13.72**	**31.07**	**22.94**	**19.75**	**57.62**	42.33	37.59	**61.78**	47.06	40.88

Table 3. **Ablation studies** on the KITTI *validation* set for the Car category.

	CM	UnC	GeP	GeU	HTL	3D@IoU=0.7			BEV@ IoU=0.7		
						Easy	Mod.	Hard	Easy	Mod.	Hard
(a)	-	-	-	-	-	15.18	11.00	9.52	21.57	16.43	13.93
(b)	✓	-	-	-	-	16.39	12.44	11.01	23.08	18.32	16.03
(c)	✓	✓	-	-	-	19.69	13.53	11.33	27.49	19.00	16.96
(d)	✓	-	✓	-	-	17.27	12.79	10.51	24.02	18.73	15.07
(e)	✓	✓	✓	-	-	18.23	13.57	11.22	26.17	19.19	16.15
(f)	✓	✓	✓	✓	-	20.86	15.70	13.21	27.54	20.80	17.77
(g)	✓	✓	✓	-	✓	21.00	15.63	12.98	30.03	21.32	18.17
(h)	✓	✓	✓	✓	✓	**22.76**	**16.46**	**13.72**	**31.07**	**22.94**	**19.75**

the competing methods on all levels of difficulty for pedestrian detection. As for cyclist detection, our approach is superior to other methods except for MonoPSR and CaDNN. The main reason is those two methods can benefit from the extra depth supervision derived from LiDAR signals, thereby improving the overall performance. In contrast, the performances of the others are limited by the few training samples (there are 14,357/2,207/734 instances in total in the KITTI train-val set). It should be noted that our method still ranks first for methods without extra data.

Latency analysis. We also test the running time of our system. We test the averaged running time on a single Nvidia TiTan XP GPU and achieve 29.4 FPS, which shows the efficiency of the inference pipeline.

4.3. Ablation Study

To understand how much improvement each component provides, we perform ablation studies on the KITTI validation set for the Car category, and the main results are summarized in Table 3.

Effectiveness of the Coordinate Maps. We concatenate a Coordinate Map (CM) for each RoI feature, and the experiment (a→b) clearly shows the effectiveness of this design, which means the location and size cues are crucial to our task. Note that the additional computing overhead introduced by CM is negligible.

Comparison of the Geometry Uncertainty Projection. We evaluate our Geometry Uncertainty Projection (GUP) module here. Note that, we think our GUP module brings gains from the following parts: geometry projection (GeP), Geometry based Uncertainty (GeU) and the Uncertainty-Confidence (UnC, Eq. 5). So we evaluate the effectiveness of these three parts respectively. First, we evaluate the effectiveness of the UnC. By comparing settings (b→c and d→e), we can find the UnC part can effectively and stably improve the overall performance, *e.g.* 1.09% improvement for (b→c) and 0.78% improvement for (d→e) on 3D detection task under moderate level. After that, we concern the GeP part effectiveness, we can see that adding GeP part improves the performance in the experiment (b→d) without UnC, but leads to an accuracy drop in the experiment (c→e) with UnC (The c and e experiments use directly learning uncertainty in the Eq. 5 to indicate confidence). This proves our motivation. It is hard for the projection-based model to directly learn accurate uncertainty and confidence because of the error amplification. Furthermore, note that the accuracy of hard cases decreases in both groups of experiments, which indicates the traditional projection cannot deal with the difficult cases caused by heavy occlusion/truncation. Second, we apply our GeU strategy based on GeP, and two groups of control experiments (e→f and g→h) are conducted. Comparing with c→e, c→f proves that our method can solve the difficulty of confidence learning in the projection-based model. The experimental results clearly demonstrate the effectiveness of our geometry modeling method for all metrics.

Table 4. Comparison with combinations of our GUP Net with some other widely used loss weights controllers on the KITTI *validation* set for the car category.

Loss weight controller	3D@IoU=0.7			BEV@ IoU=0.7		
	Easy	Mod.	Hard	Easy	Mod.	Hard
GradNorm [11]	16.19	10.49	9.04	21.80	14.74	13.02
Task Uncertainty [19]	18.95	13.94	12.18	25.07	19.45	16.74
HTL (Ours)	**22.76**	**16.46**	**13.72**	**31.07**	**22.94**	**19.75**

Influence of the Hierarchical Task Learning. We also quantify the contribution of the proposed Hierarchical Task Learning (HTL) strategy by two groups of control experiments (e→g and f→h), and both of them confirm the efficacy of the proposed HTL (improving performances for all metrics, and about 2% improvements for easy level). Also, we investigate the relationships between the loss terms and visualize the changing trend of loss weights in the training phase in Figure 7 to indicate the design effectiveness of our HTL scheme. It shows that the 2nd stage loss weights start increasing after all its tasks ({heatmap, 2D offset and 2D size}) close to convergence. And for the 3rd depth infer-

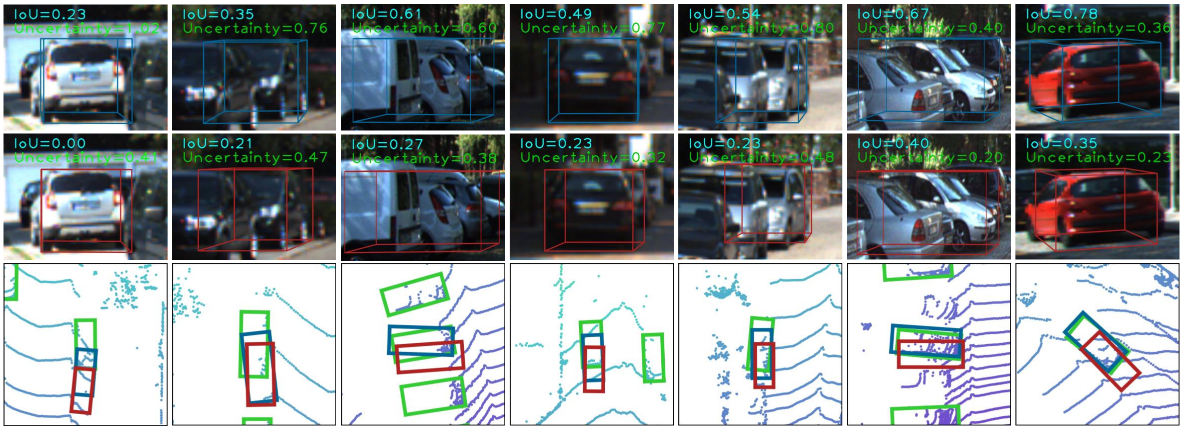

Figure 6. The visualized uncertainty examples on the validation set. The first row (Blue boxes) are results of our method. The second row (Red boxes) is the baseline results. The 3rd row shows the bird-view results (Green means the ground truth boxes). The IoU means the Intersection-over-Union between the predicted box and the corresponding ground-truth one. The uncertainty value is equal to the σ of Laplace distribution (best viewed in color.).

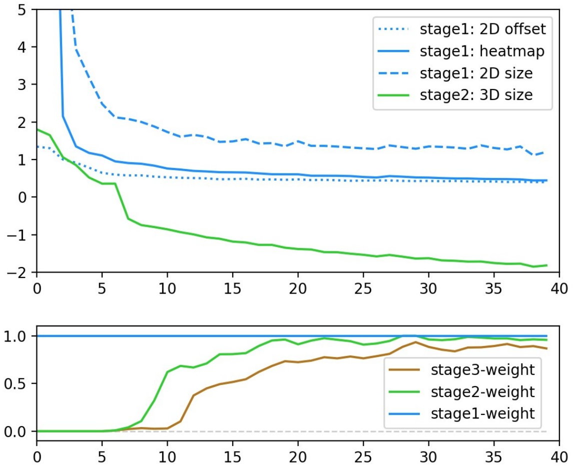

Figure 7. The upper image shows the loss curves and the bottom image means the loss weight trends. The blue, green and brown mean the 1st, 2nd and 3rd stages in the Figure 4 respectively.

ence stage, it has a similar trend. Its loss weight starts increasing at about 11th epochs. At that time, all its pre-tasks {heatmap, 2D offset and 2D size, 3D size} have achieved certain progress.

To further prove that this strategy fits our method, we also compare our HTL with some widely used loss weight controllers [11, 19] in Table 4. We can see that our methods achieve the best performance. The main reason for the poor performance of the comparison methods is that our model is a hierarchical task structure. The task-independent assumption they request does not hold in our model. And for the GardNorm, its low performance is also caused by the error amplification effect. This effect makes the magnitude of the loss function significantly change in the total training phase so it is hard for the GardNorm to balance them.

4.4. Qualitative Results

For further investigating the effectiveness of our GUP Net. We show some bad cases and corresponding uncertainties from our model and the baseline projection method (the same setting in the 4th line in Table 3). The results are shown in Figure 6. We can see that our GUP Net can predict with high uncertainties for different bad cases including occlusion and far distance. And with the improvement of the prediction results, the uncertainty prediction of our method basically decreases. And the baseline projection model gives similar low uncertainty values for that bad case, which demonstrates the efficiency of our GUP Net.

5. Conclusion

In this paper, we proposed GUP Net model for the monocular 3D object detection to tackle the error amplification ignored by conventionally geometry projection models. It combines mathematical projection priors and the deep regression power together to compute more reliable uncertainty for each object, which not only be helpful for the uncertainty-based learning but also can be used to compute the accurate confidence in the testing stage. We also proposed a Hierarchical Task Learning strategy to learn the overall model better and reduce the instability caused by the error amplification. Extensive experiments validate the superior performance of the proposed algorithm, as well as the effectiveness of each component of the model.

6. Acknowledgement

This work was supported by the Australian Research Council Grant DP200103223, FT210100228, and Australian Medical Research Future Fund MRFAI000085.

References

[1] Wentao Bao, Qi Yu, and Yu Kong. Object-aware centroid voting for monocular 3d object detection. *arXiv preprint arXiv:2007.09836*, 2020.

[2] Ivan Barabanau, Alexey Artemov, Evgeny Burnaev, and Vyacheslav Murashkin. Monocular 3d object detection via geometric reasoning on keypoints. *arXiv preprint arXiv:1905.05618*, 2019.

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

[4] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9287–9296, 2019.

[5] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *European Conference on Computer Vision*, pages 135–152. Springer, 2020.

[6] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10478–10485, 2020.

[7] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2040–2049, 2017.

[8] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12093–12102, 2020.

[11] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018.

[12] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2183–2191, 2019.

[13] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1000–1001, 2020.

[14] Xiaoqing Ye, Liang Du, Yifeng Shi, Yingying Li, Xiao Tan, Jianfeng Feng, Errui Ding, and Shilei Wen. Monocular 3d object detection via feature domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 17–34. Springer, 2020.

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[17] Tong He and Stefano Soatto. Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8409–8416, 2019.

[18] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

[19] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[20] Jason Ku, Alex D Pon, and Steven L Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11867–11876, 2019.

[21] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3559–3568, 2018.

[22] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019.

[23] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. *arXiv preprint arXiv:2001.03343*, 2, 2020.

[24] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G Narasimhan, and Jan Kautz. Neural rgb (r) d sensing: Depth and uncertainty from a video camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10986–10995, 2019.

[25] Lijie Liu, Jiwen Lu, Chunjing Xu, Qi Tian, and Jie Zhou. Deep fitting degree scoring network for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1057–1066, 2019.

[26] Lijie Liu, Chufan Wu, Jiwen Lu, Lingxi Xie, Jie Zhou, and Qi Tian. Reinforced axial refinement network for monocular 3d object detection. In *European Conference on Computer Vision*, pages 540–556. Springer, 2020.

[27] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: single-stage monocular 3d object detection via keypoint estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 996–997, 2020.

[28] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S Yu. Learning multiple tasks with multilinear relationship networks. *arXiv preprint arXiv:1506.02117*, 2015.

[29] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *European Conference on Computer Vision*, pages 311–327. Springer, 2020.

[30] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6851–6860, 2019.

[31] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2069–2078, 2019.

[32] Gregory P Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12677–12686, 2019.

[33] Pietro Morerio, Jacopo Cavazza, Riccardo Volpi, René Vidal, and Vittorio Murino. Curriculum dropout. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3544–3552, 2017.

[34] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017.

[35] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019.

[36] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8851–8858, 2019.

[37] Zengyi Qin, Jinglu Wang, and Yan Lu. Triangulation learning network: from monocular to stereo 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7615–7623, 2019.

[38] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. *arXiv preprint arXiv:2103.01100*, 2021.

[39] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018.

[40] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[41] Shaoshuai Shi, Zhe Wang, Xiaogang Wang, and Hongsheng Li. Part-A^2 net: 3d part-aware and aggregation neural network for object detection from point cloud. *arXiv preprint arXiv:1907.03670*, 2(3), 2019.

[42] Xuepeng Shi, Zhixiang Chen, and Tae-Kyun Kim. Distance-normalized unified representation for monocular 3d object detection. In *European Conference on Computer Vision*, pages 91–107. Springer, 2020.

[43] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999, 2019.

[44] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Elisa Ricci, and Peter Kontschieder. Towards generalization across depth for monocular 3d object detection. *arXiv preprint arXiv:1912.08035*, 2019.

[45] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *European Conference on Computer Vision*, pages 527–543. Springer, 2020.

[46] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019.

[47] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[48] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018.

[49] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1951–1960, 2019.

[50] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018.

[51] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11197–11206, 2020.

[52] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy:

Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018.

[53] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4106–4115, 2019.

[54] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European conference on computer vision*, pages 94–108. Springer, 2014.

[55] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[56] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4721–4730, 2021.

[57] Zhenxun Yuan, Xiao Song, Lei Bai, Zhe Wang, and Wanli Ouyang. Temporal-channel transformer for 3d lidar-based video object detection for autonomous driving. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.