2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

MEDL-U: Uncertainty-aware 3D Automatic Annotation based on Evidential Deep Learning

Helbert Paat, Qing Lian, Weilong Yao and Tong Zhang

***Abstract*— Advancements in deep learning-based 3D object detection necessitate the availability of large-scale datasets. However, this requirement introduces the challenge of manual annotation, which is often both burdensome and time-consuming. To tackle this issue, the literature has seen the emergence of several weakly supervised frameworks for 3D object detection which can automatically generate pseudo labels for unlabeled data. Nevertheless, these generated pseudo labels contain noise and are not as accurate as those labeled by humans. In this paper, we present the first approach that addresses the inherent ambiguities present in pseudo labels by introducing an Evidential Deep Learning (EDL) based uncertainty estimation framework. Specifically, we propose MEDL-U, an EDL framework based on MTrans, which not only generates pseudo labels but also quantifies the associated uncertainties. However, applying EDL to 3D object detection presents three key challenges: (1) lower pseudo label quality in comparison to other autolabelers; (2) high evidential uncertainty estimates; and (3) lack of clear interpretability and effective utilization of uncertainties for downstream tasks. We tackle these issues through the introduction of an uncertainty-aware IoU-based loss, an evidence-aware multi-task loss, and the implementation of a post-processing stage for uncertainty refinement. Our experimental results demonstrate that probabilistic detectors trained using the outputs of MEDL-U surpass deterministic detectors trained using outputs from previous 3D annotators on the KITTI val set for all difficulty levels. Moreover, MEDL-U achieves state-of-the-art results on the KITTI official *test* set compared to existing 3D automatic annotators. Code is publicly available at `https://github.com/paathelb/MEDL-U`.**

I. INTRODUCTION

Localizing 3D objects in world coordinates is a fundamental module in many robotics and autonomous driving applications. Recently, with the development of deep neural networks, network-based methods [1] have dominated this field and are capable of classifying, detecting, and reconstructing objects in 3D space.

However, the training of network-based 3D detectors requires a massive amount of data labeled with 3D bounding boxes, which often involves significant costs [2], [3]. To alleviate the heavy annotation burden, one promising direction is weakly supervised training that utilizes LiDAR data, image, and 2D bounding boxes to train a 3D object annotator [4], [5], [6], [7]. The weakly-supervised methods propose frameworks that can automatically annotate objects in 3D, minimizing the reliance on ground truth labels during downstream training of 3D detectors. Although current approaches can achieve good 3D bounding box annotations,

Helbert Paat, Qing Lian, and Tong Zhang are with the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology (HKUST), Hong Kong, China. hpaat, qlianab@connect.ust.hk, tongzhang@ust.hk. Weilong Yao is with Autowise.AI. yaoweilong@autowise.ai

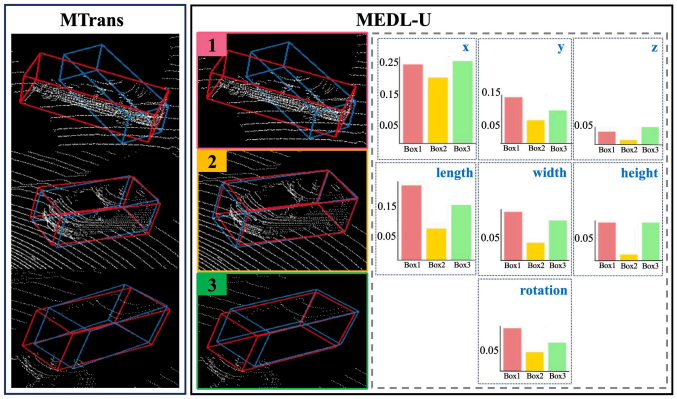

Fig. 1: Illustration of the proposed MEDL-U in comparison with current state-of-the-art 3D autolabeler, MTrans [6]. MEDL-U not only generates pseudo labels but also estimates the associated uncertainties to indicate the inaccuracy of the pseudo labels. Ground-truth boxes and pseudo labels are colored red and blue, respectively.

the generated 3D bounding boxes are not as accurate as those labeled by humans. In the illustrated pseudo labels from MTrans [6] on the left side of Figure 1, it is evident that pseudo labels 1 and 3 contain imprecise estimates of box parameters. Unfortunately, current approaches neglect these annotation noises, directly utilizing these pseudo labels to train 3D detectors. Clearly, neglecting these noises in pseudo labels degrades the effectiveness of training downstream 3D detectors. To alleviate this problem, our work considers both the tasks of annotating the 3D bounding boxes and estimating the annotation uncertainty to indicate the annotation inaccuracies. On the right side of Figure 1, we show that our work not only predicts pseudo labels but also determines the uncertainty estimates for each 3D box parameter, which are then utilized to train 3D detectors more effectively.

Evidential deep learning (EDL) has been effectively utilized for uncertainty estimation in regression tasks [8] and has found diverse applications in computer vision tasks [9], [10], [11]. Hence, we propose **M**Trans-based **E**vidential **D**eep **L**earning autolabeler with **U**ncertainty Estimation capability (MEDL-U). Our input is similar to the typical 3D automatic annotator, which comprises of a collection of scene frames, corresponding LiDAR data, 2D image, and the 2D bounding boxes for the objects. With these inputs, our goal is to develop a model that produces not only accurate 3D bounding box annotations for the surrounding objects but also a measure of uncertainty for annotated bounding box parameters (the predicted center, length, width, height, and rotation (yaw angle)), all the while avoiding any additional

manual annotation costs or huge computational overhead.

However, directly applying the EDL framework for uncertainty estimation in 3D autolabeler introduces three main challenges: (1) directly incorporating the evidential loss with evidence regularizer from EDL framework [8] results in worse performance during inference compared to the IoU-based loss as the latter unifies all 3D box parameters into one metric and aligns with the evaluation objective; (2) the uncertainties are not well-calibrated and can become unreasonably high during training; and (3) lack of interpretability, reasonable applicability and proper utilization of the generated uncertainties due to the variations in the magnitude of the evidential parameters for each 3D box.

To address these problems, we introduce an uncertainty-aware IoU loss to help the model regress high-quality box variables. Moreover, we make the multi-task loss functions evidence-regularized with the intuition that the model's predicted total evidence, as determined by the EDL framework, is inversely related to the losses for multiple tasks. Finally, we propose a post-processing step involving the rescaling of uncertainties to ensure uniformity across diverse box parameters. Simultaneously, we incorporate an objective centered on minimizing a monotonic function, denoted as f and parameterized by κ, whose primary aim is to determine the optimal value of κ that, upon applying the uncertainties through f, yields the lowest Negative Log-Likelihood (NLL) over the same limited training dataset utilized to train the 3D autolabeler.

With a limited number of annotated frames (e.g. 500 frames), our proposed MEDL-U not only generates 3D box annotations but also measures of uncertainty for each pseudo label box parameter, which can be utilized for loss reweighting during the training of existing 3D object detectors. Extensive experiments demonstrate that our MEDL-U improves the performance of 3D detectors during inference on the KITTI *val* and *test* set and outperforms previous 3D autolabelers.

II. Related Literature

A. Automatic 3D Bounding Boxes Annotation

Recently, the literature has witnessed a rise in 3D automatic annotation frameworks. An example is WSPCD [12] which allows learning 3D object parameters from a few weakly annotated examples. It has a two-stage architecture design: the first stage for cylindrical object proposal generation and the second stage for cuboids and confidence score prediction. A non-learning based approach that detects vehicles in point clouds without any 3D annotations is FGR [4] which also consists of two stages: coarse 3D segmentation stage and the bounding box estimation stage. Liu *et al.*[6] propose a Transformer-based 3D annotator called MTrans, which aims to address the prevalent sparsity problem of unstructured point clouds from LiDAR scans by generating extra 3D points through a multimodal self-attention mechanism combined with additional multi-task and self-supervision design. Different from previous appoaches, Qian *et al.*[7] propose a simplified end-to-end Transformer model, CAT, which captures local and global relationships through an encoder-decoder architecture. The encoder consists of intra-object encoder (local) and inter-object encoder (global) which performs self-attention along the sequence and batch dimensions. Additionally, several approaches (GAL [13], VS3D [14], WS3DPR [15]) have also been proposed. However, all these 3D automatic annotators only generate 3D pseudo labels without any estimation of the uncertainty or noise associated with them. In this study, we utilize the generatd 3D pseudo labels and the estimated uncertainties to train 3D detectors. This approach aims to mitigate the impact of generated noisy labels by reducing the influence of inaccurate supervision signals and enabling the model to effectively learn from more reliable pseudo labels.

B. Uncertainty Estimation and 3D Probabilistic Detection

Uncertainties in deep learning-based predictions can be categorized into two: one that is caused by inherent noise in data (aleatoric), and the other is model uncertainty due to incomplete training or model design (epistemic). As a tool for uncertainty estimation in regression tasks [8], EDL has found diverse applications in various tasks such as stereo matching [9], open set recognition [10], molecular structure prediction [16], and remote sensing [11]. In this work, we use the framework of EDL to estimate prediction uncertainties in 3D Object Detection. Utilizing these uncertainties to define pseudo label distributions, probabilistic object detectors can enable the prediction of probability distributions for object categories and bounding boxes. A framework that utilizes probabilistic detectors is GLENet [17], where the 3D detector predict distributions for the 3D box and assume that the ground truth labels follow Gaussian distribution with the uncertainties as the variance. They train the models with the KL divergence loss to supervise the predicted localization uncertainty. In this work, we follow the same approach but instead incorporate 3D pseudo label uncertainties.

III. Evidential Deep Learning (EDL) for Uncertainty Estimation in 3D Automatic Labelers

A. Automatic Annotation with Pseudo label Uncertainty Estimation

Given the point cloud data, 2D image and the 2D bounding boxes of each object, the objective of this work is to generate the 3D bounding boxes annotation for each object and estimate the corresponding uncertainty for each 3D box parameter. First, the autolabeler is initially trained with a small set of ground truth 3D bounding boxes (e.g., 500 frames of data), where the input of the autolabeler are the point cloud data, 2D image and the 2D bounding boxes of each object and the output are the estimated 3D bounding boxes and corresponding uncertainty estimates. Secondly, with the trained autolabeler, we employ it to predict the 3D bounding boxes for the remaining data and do the uncertainty estimation for the predicted 3D boxes. Finally, we leverage the predicted 3D bounding boxes and estimated uncertainty to train a downstream probabilistic 3D detector

on the massive weakly annotated data. Compared to fully supervised setting, our work only needs a few frames of labeled data to train the 3D autolabeler, which significantly reduces the manual annotation cost.

In this paper, we build the model architecture for the 3D automatic labeler from MTrans [6]. MTrans extracts object features using a multimodal self-attention module that processes point cloud and image inputs fused with point-level embedding vectors. The extracted object features are utilized for various tasks such as foreground segmentation, point generation, and 3D box regression. However, the generated 3D box pseudo labels may contain noise. To account for these inaccuracies of the pseudo labels, we incorporate uncertainty estimation task to MTrans by considering EDL, a powerful uncertainty estimation framework, and applying it to MTrans. For a straightforward incorporation of uncertainty estimation in MTrans via EDL, we include an evidential box head to regress the parameters of the evidential distribution and the 3D bounding box. For training the model, we replace the dIoU loss with the evidential loss to supervise the model in learning the box parameters. Moreover, the evidence regularizer is added to calibrate the uncertainties. However, there are problems with this approach of directly applying EDL in MTrans, which we will discuss in the next sections.

B. Background on Evidential Deep Learning

In 3D object detection, we define a 3D bounding box by its center coordinates (x, y and z), length (l), width (w), height (h), and rotation (yaw angle denoted as rot). From the viewpoint of EDL, we assume each label $j \in \mathbb{J} = \{x, y, z, l, w, h, rot\}$ is drawn i.i.d. from a Gaussian distribution where the mean μ_j and variance σ_j^2 are unknown. EDL framework assumes that μ_j is drawn from a Gaussian prior and σ_j^2 is drawn from an inverse-gamma prior.

$$j \sim \mathcal{N}(\mu_j, \sigma_j^2), \quad \mu_j \sim \mathcal{N}(\gamma_j, \sigma_j^2 \nu_j^{-1}), \quad \sigma_j^2 \sim \Gamma^{-1}(\alpha_j, \beta_j)$$

where $\gamma_j \in \mathbb{R}$, $\nu_j > 0$, and $\Gamma(\cdot)$ is the gamma function with $\alpha_j > 1$ and $\beta_j > 1$.

Let Φ_j and Θ_j denote the set of parameters $\{\mu_j, \sigma_j^2\}$ and $\{\gamma_j, \nu_j, \alpha_j, \beta_j\}$, respectively. Assuming independence of the mean and variance, the posterior $p(\Phi_j|\Theta_j)$ is defined to be an normal-inverse gamma (NIG) distribution, which is a Gaussian conjugate prior.

As presented in [8], the hyperparameters of the evidential distribution can be obtained by training a deep neural network (called evidential head) to output such values. For each 3D bounding box parameter, our model predicts four evidential parameters: $\gamma_j, \nu_j, \alpha_j, \beta_j$.

Through an analytic computation of the maximum likelihood Gaussian without repeated sampling for inference [8], EDL provides an uncertainty estimation framework in regression. We can then calculate the prediction, aleatoric, and the epistemic uncertainties for each 3D box parameter:

$$E[\mu_j] = \gamma_j, \quad E[\sigma_j^2] = \frac{\beta_j}{\alpha_j - 1}, \tag{1}$$

$$Var[\mu_j] = E[\sigma_j^2]/v_j = \frac{\beta_j}{v_j(\alpha_j - 1)}. \tag{2}$$

Maximizing the data fit: We are given the hyperparameters Θ_j of the evidential distribution as outputs of the proposed evidential head for $j \in \mathbb{J} = \{x, y, z, l, w, h, rot\}$. The likelihood of an observation y_j is computed by marginalising over the likelihood parameters Φ_j. If we choose to impose a NIG prior onto the Gaussian likelihood function results, an analytical solution is derived as follows:

$$p(y_j|\Theta_j) = St_{2\alpha_j}(y_j|\gamma_j, \frac{\beta_j(1+\nu_j)}{\nu_j \alpha_j}), \tag{3}$$

where $St_\nu(t|r,s)$ corresponds to the evaluation of the Student's t-distribution at the value t, with parameters for location (r), scale (s), and degrees of freedom (ν).

For training the EDL framework, we define the evidential loss as the mean of the negative log likelihood (NLL) for each 3D box parameter as follows:

$$\mathscr{L}_{evi}(\Psi) = -\frac{1}{|\mathbb{J}|} \sum_{j \in \mathbb{J}} \log \; p(y_j|\Theta_j). \tag{4}$$

Uncertainty Calibration: Similar to Amini *et al.*[8], we can scale the total evidence with the prediction error for each 3D box parameter in the following manner:

$$\mathscr{L}_R(\theta) = \frac{1}{|\mathbb{J}|} \sum_{j \in \mathbb{J}} \phi_j \|y_j - \gamma_j\|, \tag{5}$$

where ϕ_j is the total evidence defined as $\phi_j = 2\nu_j + \alpha_j$, and $\|\cdot\|$ is L1 norm.

C. Problems with Utilizing EDL for 3D Box Regression

- Significant variability exists in uncertainty estimates for box regression parameters, with some resulting in excessively high values. As also discussed in [18], there exist the gradient shrinkage problem in the evidential NLL loss, where the model can decrease the loss value by increasing the uncertainty instead of getting accurate box parameter estimates.
- Since the NLL-based evidential loss is not sufficient to optimize the accuracy of the prediction, the generated pseudo labels exhibit lower quality than MTrans. Empirical results also demonstrate that IoU-based loss is better than evidential loss to learn the 3D box parameters as the latter treats each 3D box parameter independently.

D. Evidence-aware Multi-task Loss

Intuitively, the loss information from the multi-task loss functions during training corresponding to the same object can be utilized to help the model understand the evidence in support to the prediction. In line with the works on uncertainty estimation [19], [9], we introduce regularized multi-task loss functions based on the form of NLL minimization of aleatoric uncertainty estimation and intuitively aligned with the learned loss attenuation. The main insight behind these loss functions is that the model's predicted evidence is inversely related to the losses for multiple tasks. Let $\mathscr{L}_t^{'}$ be the loss function corresponding to the task t where $t \in \{$seg, depth, conf, dir$\}$. The proposed evidence-aware multi-task loss is

$$\begin{aligned} \mathscr{L}_t &= \frac{\mathscr{L}_t^{'}}{1/(\phi - 1)} + \log \; \frac{1}{\phi - 1} \\ &= (\nu + 2\alpha - 1)\, \mathscr{L}_t^{'} - \log \; (\nu + 2\alpha - 1), \end{aligned} \tag{6}$$

where $\alpha = |\mathbb{J}|^{-1} \sum_{j \in \mathbb{J}} \alpha_j$ and $\nu = |\mathbb{J}|^{-1} \sum_{j \in \mathbb{J}} \nu_j$.

E. Uncertainty-aware IoU Loss

While the NLL-based evidential loss can enable the model to learn the 3D box and evidential parameters, it is not sufficient to regress 3D box variables with a quality comparable to those produced by existing autolabelers. Hence, we propose to include an IoU-based loss inspired by the DIoU loss [20] similar in form to 6. This makes $\mathscr{R}$ (penalty term for the prediction and ground truth) and IoU-related term uncertainty-aware.

$$\mathscr{L}_{IoU} = (\nu + 2\alpha - 1) \cdot (\mathscr{R} + (1 - IoU)) - \log(\nu + 2\alpha - 1). \quad (7)$$

Incorporating this new evidence-aware IoU loss improves the model's ability to handle uncertainty while generating high quality 3D box that is comparable to other 3D autolabelers. Note that the original evidence regularizer [8] could also train γ_j. Our empirical findings suggest that eliminating this regularizer is necessary as components of the proposed multi-task losses and the uncertainty-aware IoU loss already serve for regularization purposes.

F. Training of the 3D Autolabeler

In summary, the final overall loss function $\mathscr{L}$ is computed as a weighted combination of the evidential loss, the uncertainty-aware IoU loss, and the multi-task losses with evidence regularizers:

$$\begin{aligned}\mathscr{L} = \eta_{seg}\mathscr{L}_{seg} + \eta_{depth}\mathscr{L}_{depth} + \eta_{conf}\mathscr{L}_{conf} + \eta_{dir}\mathscr{L}_{dir} \\ + \eta_{evi}\mathscr{L}_{evi} + \eta_{IoU}\mathscr{L}_{IoU},\end{aligned} \quad (8)$$

where $\eta_{seg}, \eta_{depth}, \eta_{conf}, \eta_{dir}, \eta_{box}$ and η_{evi} are hyperparameters. Please refer to Figure 2 for the overall workflow.

G. Pseudo Label Uncertainty Post-processing

Prior to utilizing the pseudo labels and uncertainties as supervision signals in the training process of existing 3D detectors, it is necessary to apply a post-processing step to address the variability in the magnitudes of 3D box parameter uncertainties and to make the uncertainties more appropriate for downstream task. We propose a post-processing procedure that rescales the uncertainties but ensures that the resultant uncertainty values maintain its Spearman's rank correlation with two crucial metrics: the L2 norm of the residuals and the 3D IoU between the predicted box and the ground truth box. Initially, the predicted epistemic uncertainty for each 3D box parameter j, where $j \in \mathbb{J} = \{x, y, z, l, w, h, rot\}$, undergoes a transformation which constrains these uncertainties within a specific range through the application of a simple monotonic function. Moreover, we pass uncertainty estimate for each box parameter x_j to a function f, formulated as $f(x_j) = x_j^{1/\kappa_j}$, where we carefully select values of κ_j to minimize the NLL of the uncertainties for each 3D box parameter j with respect to the same limited training data used to train the 3D autolabeler. The assumption is that lower NLL means better uncertainty estimates, consequently improving supervision for the downstream task. Lastly, we generate multiple sets of uncertainties by passing them to the function $g(x_j) = x_j^{1/\varepsilon_j}$, with ε_j acting as a downstream training hyperparameter. Because κ_j parameter only effectively minimizes NLL over the limited labeled training data, we introduce ε_j to appropriately adjust the uncertainties and achieve lower NLL over the entire training dataset.

H. Downstream Training via Probabilistic 3D Detector

A probabilistic object detector enables the inclusion of pseudo label uncertainties during the training phase, where these uncertainties can be interpreted as factors for reweighting losses. Our work follows [21], [17] in transforming a 3D detector from deterministic to probabilistic, where the detection head is enforced to estimate a Gaussian probability distribution over bounding boxes. Let θ indicate the learnable network weights of a typical detector, $\hat{y}$ indicate the predicted box parameters, and $\hat{\sigma}^2$ the predicted localization variance. Moreover, the pseudo ground truth bounding boxes are also assumed to be Gaussian distributed having a variance σ^2 where σ^2 are estimated by MEDL-U. Let the pseudo ground truth bounding box be denoted by y_g and D refer to the pseudo label distribution. Hence, the generated pseudo label uncertainty can be incorporated in the KL Divergence loss between the distribution of prediction and pseudo ground truth in the detection head:

$$\begin{aligned}L_{reg} &= D_{KL}(P_D(y) || P_\theta(y)) \\ &= log\frac{\hat{\sigma}}{\sigma} + \frac{\sigma^2}{2\hat{\sigma}^2} + \frac{(y_g - \hat{y})^2}{2\hat{\sigma}^2}.\end{aligned} \quad (9)$$

Similar to [17], [22], we also employ 3D Variance Voting which uses the predicted variance $\hat{\sigma}^2$ to combine nearby bounding boxes for better 3D localization prediction.

IV. Experimental Setup

A. Dataset

The KITTI Object Detection dataset [23], renowned for 3D detection in autonomous driving, is employed in this study. The dataset has a total of 7481 frames with labels in 3D. Following the official procedure, the dataset is divided into training and validation sets, consisting of 3,712 and 3,769 frames, respectively. Similar to previous works [4], [5], [6], [7], we concentrate on the Car class and exclude objects with fewer than 5 foreground LiDAR points.

B. Implementation Details and Model Structure

Our method is implemented in PyTorch [24]. Similar to the original MTrans [6], MEDL-U architecture incorporates four multimodal self-attention layers, each having a hidden size of 768 and 12 attention heads. Unless otherwise stated, training the autolabeler requires 500 annotated frames only. We employed a dropout rate of 0.4 and utilized the Adam optimizer with a learning rate of $0.60e-04$. Autolabeler training is conducted for 300 epochs, with a batch size of 5. Training of probabilistic 3D detectors is conducted for 80 epochs. Note that only epistemic uncertainties from MEDL-U are utilized. Unless specified differently, hyperparameter tuning on the KITTI validation set suggests using $\varepsilon = 1$ for PointPillars and CIA-SSD, and $\varepsilon = 5$ for other detectors. All trainings are executed on NVIDIA RTX 2080Ti GPU.

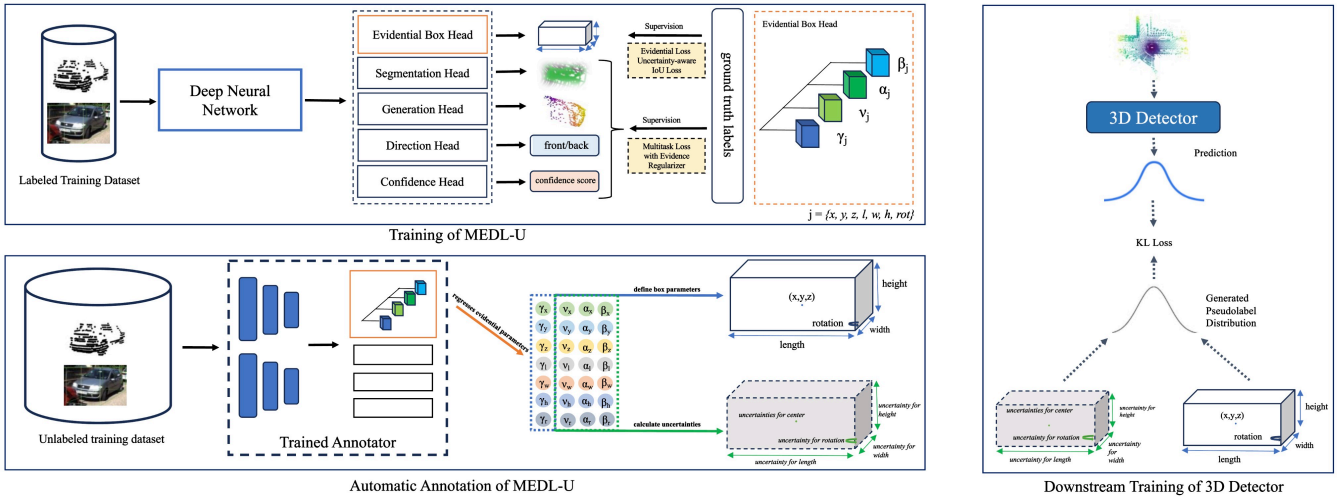

Fig. 2: Architecture of the Training and Automatic Annotation Workflow of MEDL-U. The evidential box head regresses the evidential parameters which can be used to calculate the 3D box parameters and the uncertainties. During the automatic annotation, MEDL-U regresses 3D box parameters and the associated uncertainties for the unlabeled data. In the downstream training of probabilistic 3D detectors, the generated pseudo labels provide supervision during training and the associated box parameter uncertainties serve as factors for reweighting via the KLD loss.

MEDL-U evidential regression head consists of four output units for each of the seven box attributes. The input to this head are the transformed element representations extracted from the self-attention layer. The evidential regression head comprises a sequence of linear layers, followed by LayerNorm, a Dropout layer, and a ReLU activation function. To ensure that certain values are positive, a Softplus activation is applied to ν, α, and β, where α is then incremented by 1 to ensure $\alpha > 1$. For γ, a linear activation is used. Overall, MEDL-U has over 23 million trainable parameters. It is trained to learn 3D box parameters evidential distributions, along with the other tasks of segmentation, point generation, direction, and confidence prediction. In the next sections, MEDL refers to utilizing the pseudo labels only while MEDL-U refers to utilizing both pseudo labels and the uncertainties in the downstream training of 3D detectors.

C. Evaluation metrics

1) 3D Box Prediction: To assess localization, we measure the Average Precision, specifically for 3D objects (AP_{3D}) and Birds Eye View (BEV) with a stringent IoU threshold of 0.70 to determine positive detections. Average Precision at 40 points (R40) means that the precision and recall are calculated at 40 different recall levels.

2) Uncertainty Estimation: Widely employed in previous studies [25], the Negative Log-Likelihood (NLL) is a metric that evaluates the model's ability to estimate uncertainty. Lower NLL values indicate more accurate uncertainty estimation. Moreover, we also calculate the Spearman's rank correlation coefficients of the predicted uncertainties to the corresponding L2 norm of the residuals.

D. Experiment on Different Kinds of 3D Detectors

We evaluate several one-stage and two-stage 3D detectors on the KITTI *val* set when trained using outputs from various annotators. As seen in Table I, detectors trained on MEDL-U outputs outperform vanilla deterministic 3D detectors trained on MTrans and MEDL pseudo labels, demonstrating the effectiveness of utilizing not only the pseudo labels but also the uncertainty estimates for each 3D box parameter.

E. Comparison with 3D Automatic Annotation Frameworks

As shown in Table II, evaluating the probabilistic PointRCNN trained with MEDL-U ouputs on the KITTI *val* set yields superior performance relative to all existing and current 3D autolabeling methods in terms of AP_{3D}.

In Table III, probabilistic PointRCNN trained with outputs of MEDL-U on the entire KITTI training and val sets results in better performance on the KITTI official *test* set compared to PointRCNN trained with vanilla MTrans. Moreover, PointRCNN trained with MEDL-U outputs yields superior performance in terms of AP_{3D} and AP_{BEV} for both Easy and Moderate levels relative to all existing 3D automatic labeling method. MEDL-U does not outperform CAT across all difficulty levels, which is understandable considering that MEDL-U is built upon MTrans, chosen for its open-source availability. Moreover, CAT is trained for 1000 epochs and a batch size of 24, which is different from the training setting for MTrans and MEDL-U. We argue that the enhancements seen in MEDL-U over MTrans can also be applied to CAT.

We also show evaluation performance on the KITTI *val*

TABLE I: Comparison of AP_{3D} $R40$ on the KITTI *val* set using 3D detectors trained on MTrans pseudo labels, MEDL pseudo labels only, and MEDL-U pseudo labels and uncertainties. Results are produced from our own experiments.

	Easy			Moderate			Hard		
Detector	MTrans	MEDL	MEDL-U	MTrans	MEDL	MEDL-U	MTrans	MEDL	MEDL-U
SECOND [26]	90.69	89.53	**90.95**	79.63	80.53	**82.69**	76.11	76.20	**77.82**
PointPillars [27]	86.70	87.34	**91.16**	75.36	78.07	**80.44**	72.06	74.45	**75.43**
CIA-SSD [28]	89.81	90.50	**92.33**	78.52	80.79	**83.27**	73.22	75.48	**78.30**
Voxel RCNN [29]	92.25	92.30	**92.59**	83.25	82.95	**83.72**	80.11	79.92	**80.73**
PointRCNN [30]	91.67	91.56	**92.47**	80.73	80.45	**81.59**	75.75	77.54	**78.68**

TABLE II: AP_{3D} Results on KITTI *val* set, compared to the fully supervised PointRCNN and other weakly supervised baselines. Results here are from the official published results.

Method	Reference	Full Supervision	Easy	Moderate	Hard
PointRCNN [30]	CVPR 2019	✓	88.99	78.71	78.21
WS3D [31]	ECCV 2020	BEV Centroid	84.04	75.10	73.29
WS3D (2021) [12]	TPAMI 2022	BEV Centroid	85.04	75.94	74.38
FGR [4]	ICRA 2021	2D Box	86.68	73.55	67.91
MAP-Gen [5]	ICPR 2022	2D Box	87.87	77.98	76.18
MTrans [6]	ECCV 2022	2D Box	88.72	78.84	77.43
CAT [7]	AAAI 2023	2D Box	89.19	79.02	77.74
MEDL (Ours)	-	2D Box	89.07	78.68	76.99
MEDL-U (Ours)	-	2D Box	**89.26**	**79.27**	**78.05**

TABLE III: Results of KITTI official *test* set, compared to the fully supervised PointRCNN and other weakly supervised baselines. Results here are from official published results.

		AP_{3D}			AP_{BEV}		
Method	Modality	Easy	Moderate	Hard	Easy	Moderate	Hard
PointRCNN [30]	LiDAR	86.96	75.64	70.70	92.13	87.39	82.72
WS3D [31]	LiDAR	80.15	75.22	70.05	90.11	84.02	76.97
WS3D (2021) [12]	LiDAR	80.99	70.59	64.23	90.96	84.93	77.96
FGR [4]	LiDAR	80.26	68.47	61.57	90.64	82.67	75.46
MAP-Gen [5]	LiDAR+RGB	81.51	74.14	67.55	90.61	85.91	80.58
MTrans [6]	LiDAR+RGB	83.42	75.07	68.26	91.42	85.96	78.82
CAT [7]	LiDAR	84.84	75.22	**70.05**	91.48	85.97	**80.93**
MEDL-U (Ours)	LiDAR+RGB	**85.49**	**75.96**	69.12	**91.86**	**86.68**	79.44

TABLE IV: AP_{3D} Results on KITTI *val* and official *test* set using PointPillars when the 3D autolabelers are trained using 500 and 125 frames of annotated data. We produce KITTI *test* results for MTrans + PointPillars (125f), while other results on MTrans are officially published results.

		KITTI *val* set			KITTI official *test* set		
Method	Supervision	Easy	Moderate	Hard	Easy	Moderate	Hard
MTrans + PointPillars	500f	86.69	76.56	72.38	77.65	67.48	62.38
MEDL-U + PointPillars	500f	**88.12**	**78.24**	**76.58**	**82.77**	**72.74**	**65.57**
MTrans + PointPillars	125f	83.70	71.66	66.67	71.49	59.04	51.79
MEDL-U + PointPillars	125f	**85.49**	**74.66**	**66.95**	**78.13**	**66.46**	**57.40**

and *test* set using PointPillars when MTrans and MEDL-U are trained with 500 and 125 annotated frames. MEDL-U significantly improves the baseline as shown in Table IV.

F. Comparison with Other Uncertainty Estimation Methods

Using 3D box parameter uncertainties generated by MEDL-U and other popular uncertainty estimation methods, we evaluate probabilistic version of PointRCNN on the KITTI *val* set. Three baseline methods were implemented: (1) A Monte Carlo dropout (MC Dropout) system with a dropout rate of 0.2 and was forwarded 5 times during inference. (2) A Deep Ensemble of 5 systems trained with different random seeds. (3) Confidence score predicted by vanilla MTrans was used as proxy to uncertainty. As shown in Table V, PointRCNN trained with MEDL-U ($\varepsilon = 5$) yields the overall best result in terms of $AP_{3D}R40$. Noticeably, using other uncertainty estimation methods to generate uncertainties also effectively increase $AP_{3D}R40$ of the *base* method, although MC Dropout and Deep Ensemble come at the cost of huge additional computational overhead. While the Deep Ensemble approach can improve the *base* result, the original pseudo label quality is relatively poor.

TABLE V: Comparison of AP_{3D} $R40$ on KITTI *val* set when PointRCNN is trained using outputs from different uncertainty estimation methods. *base* refers to utilizing pseudo labels only in training PointRCNN.

Method	Easy	Moderate	Hard
vanilla MTrans [6]	91.67	80.73	75.75
conf ($\varepsilon = 4$)	92.10	80.87	75.91
conf ($\varepsilon = 5$)	92.59	81.42	76.41
MC Dropout (*base*)	92.14	80.43	75.76
MC Dropout ($\varepsilon = 4$)	92.50	81.22	76.44
MC Dropout (ε = 5)	**92.91**	81.35	76.50
Deep Ensemble (*base*)	84.47	74.99	72.14
Deep Ensemble ($\varepsilon = 4$)	85.06	75.99	71.18
Deep Ensemble ($\varepsilon = 5$)	84.78	75.49	72.82
MEDL (*base*)	91.56	80.45	77.54
MEDL-U ($\varepsilon = 4$)	91.73	81.11	78.27
MEDL-U (ε = 5)	92.47	**81.59**	**78.68**

TABLE VI: Evaluating uncertainties using NLL and Spearman's rank correlation with residuals. Values in bold achieve the best score. Underlined values achieve the second best.

	Negative Log Likelihood (NLL)							Correlation Coefficient (in %)						
	x	y	z	l	w	h	rot	x	y	z	l	w	h	rot
Confidence score	4.5	2.4	**-1.9**	3.6	1.4	**-1.3**	5.3	40	50	33	**44**	43	35	25
MC Dropout	2.2	1.7	-1.3	1.9	1.4	-0.6	5.5	39	45	45	-2	**45**	21	24
Deep Ensemble	4.4	**0.1**	-1.5	**0.6**	**-1.2**	-0.7	**0.9**	**49**	55	**49**	41	37	39	31
MEDL-U	**1.9**	0.4	-1.8	1.0	-1.0	-1.2	1.3	**49**	**56**	43	39	39	**40**	**61**

TABLE VII: Ablation Results showing Spearman's rank correlation (in %) between generated uncertainties and the L2 norm of the residuals.

Model Design	x	y	z	l	w	h	rot
Replace original IoU loss with $\mathscr{L}_{evi}$ & $\mathscr{L}_R$	45.8	54.8	41.2	35.6	35.2	39.9	59.3
Include evidence-aware multi-task loss $\mathscr{L}_t$	47.7	55.4	42.3	36.1	36.7	39.7	59.8
Replace $\mathscr{L}_R$ with uncertainty-aware IoU loss	**49.0**	**55.8**	**42.8**	**38.7**	**38.6**	**39.9**	**60.8**

TABLE VIII: Comparison of the AP_{3D} $R40$ when training Voxel R-CNN using different values of ε.

	Voxel R-CNN + MEDL-U					
$AP_{3D}R40$	1	2	3	4	5	6
Easy	92.32	92.15	92.15	92.16	**92.59**	92.15
Moderate	83.54	83.35	83.50	83.48	**83.72**	83.51
Hard	78.65	80.39	80.59	80.60	**80.73**	80.52

G. Uncertainty Evaluation and Ablation Results

Table VI shows that the uncertainties generated by MEDL-U achieve the overall best result in terms of NLL and Spearman's rank correlation coefficient. Specifically, it achieves the highest uncertainty Spearman's rank correlation with residuals for the x and y coordinate of the center, height, and rotation, and consistently among the two lowest calculated NLL for all the 3D box parameters. Details on ablation results and results on varying ε for the downstream task can be seen in Tables VII and VIII, respectively.

V. Conclusion

In this paper, we propose MEDL-U, a 3D automatic labeler with EDL framework that generates not only high-quality pseudo labels but also uncertainties associated with each 3D box parameter. Compared with previous autolabeling approaches, our method achieves overall better results in the downstream 3D object detection task on the KITTI *val* and *test* set, showing the importance of quantifying uncertainties or noise in pseudo labels.

REFERENCES

[1] R. Qian, X. Lai, and X. Li, "3d object detection for autonomous driving: A survey," *ArXiv*, vol. abs/2106.10823, 2021.

[2] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 567–576, 2015.

[3] P. Wang, X. Huang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 2702–2719, 2018.

[4] Y. Wei, S.-C. Su, J. Lu, and J. Zhou, "Fgr: Frustum-aware geometric reasoning for weakly supervised 3d vehicle detection," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4348–4354, 2021.

[5] C. Liu, X. Qian, X. Qi, E. Y. Lam, S.-C. Tan, and N. Wong, "Map-gen: An automated 3d-box annotation flow with multimodal attention point generator," *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 1148–1155, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247779303

[6] C. Liu, X. Qian, B. Huang, X. Qi, E. Y. Lam, S.-C. Tan, and N. Wong, "Multimodal transformer for automatic 3d annotation and object detection," in *European Conference on Computer Vision*, 2022.

[7] X. Qian, C. Liu, X. Qi, S.-C. Tan, E. Y. Lam, and N. Wong, "Context-aware transformer for 3d point cloud automatic annotation," in *AAAI Conference on Artificial Intelligence*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257766591

[8] A. Amini, W. Schwarting, A. P. Soleimany, and D. Rus, "Deep evidential regression," *ArXiv*, vol. abs/1910.02600, 2019.

[9] C. Wang, X. Wang, J. Zhang, L. Zhang, X. Bai, X. Ning, J. Zhou, and E. R. Hancock, "Uncertainty estimation for stereo matching based on evidential deep learning," *Pattern Recognit.*, vol. 124, p. 108498, 2021.

[10] W. Bao, Q. Yu, and Y. Kong, "Evidential deep learning for open set action recognition," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13 329–13 338, 2021.

[11] J. Gawlikowski, S. Saha, A. M. Kruspe, and X. X. Zhu, "An advanced dirichlet prior network for out-of-distribution detection in remote sensing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, pp. 1–1, 2022.

[12] Q. Meng, W. Wang, T. Zhou, J. Shen, Y. Jia, and L. V. Gool, "Towards a weakly supervised framework for 3d point cloud object detection and annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 4454–4468, 2021.

[13] D. Yin, H. Yu, N. Liu, F. Yao, Q. He, J. Li, Y. Yang, S. Yan, and X. Sun, "Gal: Graph-induced adaptive learning for weakly supervised 3d object detection," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[14] Z. Qin, J. Wang, and Y. Lu, "Weakly supervised 3d object detection from point clouds," *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

[15] H. Liu, H. Ma, Y. Wang, B. Zou, T. Hu, R. Wang, and J. Chen, "Eliminating spatial ambiguity for weakly supervised 3d object detection without spatial labels," *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.

[16] A. P. Soleimany, A. Amini, S. Goldman, D. Rus, S. N. Bhatia, and C. W. Coley, "Evidential deep learning for guided molecular property prediction and discovery," *ACS Central Science*, vol. 7, pp. 1356 – 1367, 2021.

[17] Y. Zhang, Q. Zhang, Z. Zhu, J. Hou, and Y. Yuan, "Glenet: Boosting 3d object detectors with generative label uncertainty estimation," *ArXiv*, vol. abs/2207.02466, 2022.

[18] D. Oh and B. Shin, "Improving evidential deep learning via multi-task learning," in *AAAI Conference on Artificial Intelligence*, 2021.

[19] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *ArXiv*, vol. abs/1703.04977, 2017.

[20] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *AAAI Conference on Artificial Intelligence*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:208158250

[21] D. Feng, Z. Wang, Y. Zhou, L. Rosenbaum, F. Timm, K. C. J. Dietmayer, M. Tomizuka, and W. Zhan, "Labels are not perfect: Inferring spatial uncertainty in object detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 9981–9994, 2020.

[22] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2883–2892, 2018.

[23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Neural Information Processing Systems*, 2019.

[25] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *ArXiv*, vol. abs/1506.02142, 2015.

[26] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors (Basel, Switzerland)*, vol. 18, 2018.

[27] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 689–12 697, 2018.

[28] W. Zheng, W. Tang, S. Chen, L. Jiang, and C.-W. Fu, "Cia-ssd: Confident iou-aware single-stage object detector from point cloud," in *AAAI Conference on Artificial Intelligence*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:227335169

[29] J. Deng, S. Shi, P.-C. Li, W. gang Zhou, Y. Zhang, and H. Li, "Voxel r-cnn: Towards high performance voxel-based 3d object detection," *ArXiv*, vol. abs/2012.15712, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229923684

[30] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–779, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:54607410

[31] Q. Meng, W. Wang, T. Zhou, J. Shen, L. V. Gool, and D. Dai, "Weakly supervised 3d object detection from lidar point cloud," in *European Conference on Computer Vision*, 2020.