Autoregressive Uncertainty Modeling for 3D Bounding Box Prediction

YuXuan Liu[1,2], Nikhil Mishra[1,2], Maximilian Sieb[1], Yide Shentu[1,2], Pieter Abbeel[1,2], and Xi Chen[1]

[1] Covariant.ai
[2] UC Berkeley

Abstract. 3D bounding boxes are a widespread intermediate representation in many computer vision applications. However, predicting them is a challenging task, largely due to partial observability, which motivates the need for a strong sense of uncertainty. While many recent methods have explored better architectures for consuming sparse and unstructured point cloud data, we hypothesize that there is room for improvement in the modeling of the output distribution and explore how this can be achieved using an autoregressive prediction head. Additionally, we release a simulated dataset, COB-3D, which highlights new types of ambiguity that arise in real-world robotics applications, where 3D bounding box prediction has largely been underexplored. We propose methods for leveraging our autoregressive model to make high confidence predictions and meaningful uncertainty measures, achieving strong results on SUN-RGBD, Scannet, KITTI, and our new dataset[3].

Keywords: 3D bounding boxes, 3D bounding box estimation, 3D object detection, autoregressive models, uncertainty modeling

1 Introduction

Predicting 3D bounding boxes is a core part of the computer vision stack in many real world applications, including autonomous driving, robotics, and augmented reality. The inputs to a 3D bounding box predictor usually consist of an RGB image and a point cloud; the latter is typically obtained from a 3D sensor such as LIDAR or stereo depth cameras. These 3D sensing modalities have their own idiosyncrasies: LIDAR tends to be accurate but very sparse, and stereo depth can be both sparse and noisy. When combined with the fact that objects are only seen from one perspective, the bounding-box prediction problem is fundamentally underspecified: the available information is not sufficient to unambiguously perform the task.

Imagine that a robot is going to grasp an object and manipulate it — understanding the uncertainty over the size can have a profound impact on what the robot decides to do next. For example, if it uses the predicted bounding box

[3] Code and dataset are available at bbox.yuxuanliu.com

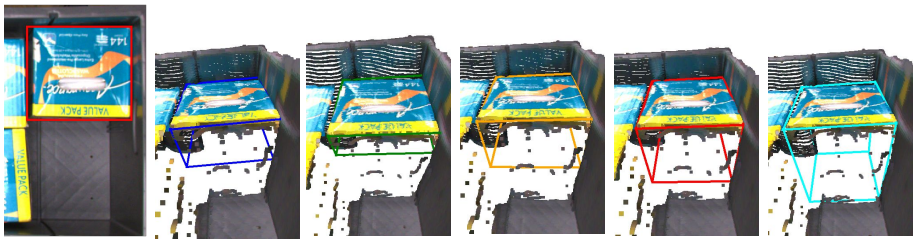

Fig. 1: a) In this scene from a real-world robotics application, how tall is the object highlighted in red? b) A pointwise model could output only one box prediction with no notion of uncertainty c)-e) Predictions from our confidence box method. Notice that the predicted box expands in the direction of uncertainty as we increase the confidence requirement. f) Our dimension conditioning method can leverage additional information to make more accurate predictions.

to avoid collisions during motion planning, then we may want to be conservative and err on the larger side. However, if it is trying to pack the items into a shipment, then having accurate dimensions may also be important.

Consider the scene depicted in Figure 1, which we observed in a real-world robotics application. From the image of the object in a), it is fairly easy to gauge the width and length of the indicated object, but how tall is it? The object could be as deep as the bin, or it could be a stack of two identical objects, or even a thin object – but from the available information, it is impossible to say for sure. Formulating bounding box prediction as a regression problem results in a model that can only make a "pointwise" prediction – even in the face of ambiguity, we will only get a single predicted bounding box, shown in b).

A sufficiently expressive bounding-box model should be able to output the entire range of plausible bounding box hypotheses and make different predictions for different confidence requirements. A 0.5-confidence box d) must contain the object 50% of the time while a 0.8-confidence box e) will expand in the direction of uncertainty to contain the object 80% of the time. Moreover, such a model could leverage additional information, such as known dimensions of an object, to make even more accurate predictions, as shown in f).

Setting aside partial observability, the prediction space has complexities that require care in the design of a bounding-box estimator. Making accurate predictions requires the estimator to reason about rotations, which has been observed to be notoriously difficult for neural networks to predict and model uncertainty over [29,5,16]. Many existing methods sidestep this problem by constraining their predictions to allow rotation about a single axis or no rotations at all. This can be sufficient for some applications but has shortcomings for the general case.

A common thread that links these challenges together is the necessity to reason about uncertainty. This has been largely underexplored in existing work, but we hypothesize that it is critical to improving 3D bounding box estimators

and expanding their usability in applications of interest. We propose to tackle this problem by predicting a more expressive probability distribution that explicitly accounts for the relationships between different box parameters. Using a technique that has proven effective in other domains, we propose to model 3D bounding boxes autoregressively: that is, to predict each box component sequentially, conditioned on the previous ones. This allows us to model multimodal uncertainty due to incomplete information, make high confidence predictions in the face of uncertainty, and seamlessly relax the orientation constraints that are popular in existing methods. To summarize our contributions:

1. We propose an autoregressive formulation to 3D bounding box prediction that can model complex, multimodal uncertainty. We show how this formulation can gracefully scale to predict complete 3D orientations, rather than the 0- or 1-D alternatives that are common in prior work.
2. We propose a method to make high confidence predictions in ambiguous scenarios and estimate useful measures of uncertainty.
3. We introduce a simulated dataset of robotics scenes that illustrates why capturing uncertainty is important for 3D bounding box prediction, as well as the benefits and challenges of predicting full 3D rotations.
4. We show that our formulation applies to both traditional 3D bounding box estimation and 3D object detection, achieving competitive results on popular indoor and autonomous driving datasets in addition to our dataset.

2 Related Work

3D Bounding-box Estimation: Early work on 3D bounding box prediction [14,19] assumes that object detection or segmentation has already been performed, and the bounding box predictor solely needs to identify a single 3D bounding box within a filtered point cloud. In this paper, we refer to this task as *3D bounding-box estimation*. Much of this work focused on developing architectures to easily consume point cloud data, which often can be sparse and/or unstructured when obtained from real-world data.

3D Object Detection: Recently, a number of methods [21,20,9,13,27,22,17] have explored how to jointly perform object detection and 3D bounding box estimation, rather than treating them as two explicit steps. This task is known as *3D object detection* and is quickly gaining popularity over the decoupled detection and estimation tasks. The main focus is on how to take the network architectures that have proven successful at the estimation task (which have strong inductive biases for operating on point clouds), and combine them with the architectures commonly used for the 2D object detection method (which are usually based on region proposals).

Uncertainty Modeling in Object Detection: Uncertainty modeling has been studied in the context of 2D and 3D Object Detection [11,8,28,12,6]. In many cases, these methods will use independent distributions, such as Gaussian

or Laplace, to model uncertainty over box parameters such as corners, dimensions, and centers [7,11,1]. While these distributions may capture some uncertainty for simple box parameterizations, they don't capture correlations across parameters and have yet to be proven on full 3D rotations.

Autoregressive Models: Deep autoregressive models are frequently employed across a variety of domains. In deep learning, they first gained popularity for generative modeling of images [25,15,26], since they can model long-range dependencies to ensure that pixels later in the autoregressive ordering are sampled consistently with the ones sampled earlier. In addition to being applied to other high-dimensional data such as audio [15], they have also been shown to offer precise predictions even for much lower-dimensional data, such as robot joint angles or motor torques [10].

3 Autoregressive 3D Bounding Box Prediction

3D bounding box estimation is typically formulated as a regression problem over the dimensions $d = (d_x, d_y, d_z)$, center $c = (c_x, c_y, c_z)$, and rotation $R = (\psi, \theta, \phi)$ of a bounding box, given some perceptual features h computed from the scene, e.g. from an image and point cloud. Prior work has explored various parametrizations and loss functions, but a notable salient feature to observe is that they all predict a *pointwise* estimate of the bounding box: the model simply outputs all of the box parameters at once. In 3D object detection, such regression is typically applied to every box within a set of candidates (or *anchors*), and fits into a larger cascade that includes classifying which anchors are relevant and filtering out unnecessary or duplicate anchors. In practice, this formulation can be greatly limiting, especially in the face of partial observability or symmetry.

3.1 Autoregressive Modeling

We propose to tackle this problem by autoregressively modeling the components of a 3D bounding box. That is, for some ordering of the components (e.g. dimensions $\rightarrow$ center $\rightarrow$ orientation, or any permutation thereof), such a predictor will sequentially predict each component conditioned on the previous ones. In theory, the particular autoregressive ordering should not matter; empirically, we find that dimensions $\rightarrow$ center $\rightarrow$ orientation was effective, so we use this ordering for our model. Having dimension as first in the autoregressive ordering also enables us to condition on dimensions when they are known which can be effective at improving the prediction accuracy.

We discretize the box parameters rather than predicting continuous values, which is a well-known technique that allows the model to easily express multimodal distributions [25]. For rotations, we chose Euler angles since each dimension has a fixed range and does not to be normalized. To make discrete dimension and center predictions, we normalize those parameters so that they can fit within a fixed set of discrete bins. We normalize dimensions by some scale s so that most values of d/s are within the range $[0, 1]$, and offset the centers by

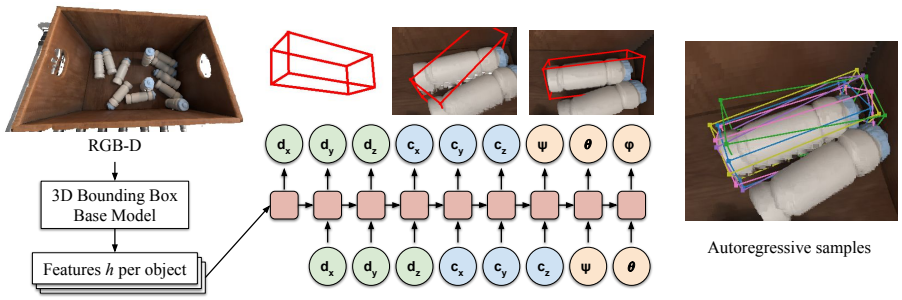

Fig. 2: We compute per-object features h using a base model from RGB-D input. Then, we autoregressively sample dimensions, center, and rotations, each step conditioned on the previous one. We can express uncertainty through samples, such as the rotational symmetry of the bottle, whereas pointwise models could only make a single prediction.

c_0 so that most normalized centers $(c - c_0)/s$ are within the range $[-1, 1]$. We use 512 bins for each dimension and adjust the bin range to achieve on average ≥ 0.99 IOU with the quantized box and $< 0.1\%$ overflow or underflow due to quantization.

From RGB-D inputs we extract a fixed-dimensional feature vector h for each object. For each parameter $b = (d_x, d_y, d_z, c_x, c_y, c_z, \psi, \theta, \phi)$ in the autoregressive ordering, we model $p(b_i|b_1, \ldots, b_{i-1}, h)$ using a MLP with 2-3 hidden layers. This autoregressive model is then trained using maximum likelihood:

$$\log p(b|h) = \sum_{i=1}^{9} \log p(b_i|b_1, \ldots, b_{i-1}, h) \tag{1}$$

3.2 Model Architectures

Our autoregressive prediction scheme can be applied to any type of 3D bounding box predictor. In this section, we discuss how it might be applied in two different contexts: 3D object detection and 3D bounding box estimation.

Autoregressive 3D Object Detection. FCAF3D [20] is a state-of-the-art 3D object detection method that was heavily engineered to exploit sparse and unstructured point clouds. Given a colored point cloud, it applies a specialized feature extractor consisting of sparse 3D convolutions, and then proposes 3D bounding boxes following a popular single-stage detector, FCOS [24].

Autoregressive FCAF3D: We can make FCAF3D autoregressive by adding a head and training this head with maximum likelihood in addition to the FCAF3D loss $L_F(h, y)$ (Figure 3). We found that the pointwise box prediction was useful to condition the autoregressive prediction and estimate the scaling

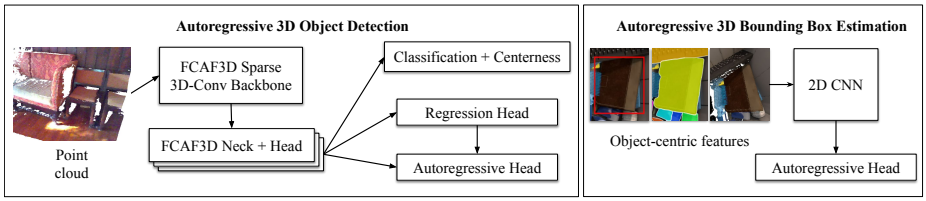

Fig. 3: For indoor 3D Object Detection, we use FCAF3D as a base model with an autoregressive head for bounding box prediction. For 3D Bounding Box Estimation we take object-centric features from a 2D object detector and pass them into a 2D CNN for autoregressive bounding box prediction.

normalization factor $s = \max\{d'_x, d'_y, d'_z\}$, where d' is the pointwise dimension prediction of FCAF3D. Bounding box centers c are normalized by the output locations c_0 of the sparse convolutions and scaled by the same s: $(c - c_0)/s$. Since 3D object detection datasets have at most one degree of freedom for rotation, we predict only one θ parameter for box rotation.

To optimize the autoregressive prediction for higher IOU, we sample boxes $b \sim p(b|h)$ and maximize the IOUs of the samples with the ground truth box y. For this optimization, we use the conditional expectation b' where $b'_i = \mathbb{E}[b_i|b_1, \ldots, b_{i-1}, h]$ (since b' is differentiable) to maximize $IOU(b', y)$. Altogether, we train autoregressive FCAF3D using the combined loss:

$$L(h, y) = L_F(h, y) - \log p(b|h) + \mathbb{E}_{b \sim p(b|h)}[1 - IOU(b', y)] \tag{2}$$

Autoregressive PV-RCNN: Lidar-based object detection networks, such as PV-RCNN [22], typically have different architectures and inductive biases than indoor detection models. However, we show that our autoregressive box parameterization is agnostic to the underlying architecture by applying it to PV-RCNN. We propose Autoregressive PV-RCNN by extending the proposal refinement head to be autoregressive, modeling the residual Δr^α as discrete autoregressive $p(\Delta r^\alpha|h)$. Then, we add $-\log p(\Delta r^\alpha|h)$ to the total training loss.

Autoregressive 3D Bounding Box Estimation. 3D Bounding Box Estimation assumes that object detection has already been performed in 2D, and we simply need to predict a 3D bounding box for each detected object. To highlight that our autoregressive prediction scheme can be applied to any bounding box predictor, we chose a model architecture that is substantially different from FCAF3D. For each detected object, we take an object-centric crop of the point cloud, normals, and object mask as input to a 2D-CNN, producing a fixed feature vector h per object. This h is used as features for our autoregressive parameterization $p(b|h)$. See Appendix A for more details on the architecture.

To normalize the input and box parameters, we scale by the range of the first and third quartiles of each point cloud dimension $s = Q_3 - Q_1$, and recenter by the mean of the quartiles $c_0 = \frac{Q_1+Q_3}{2}$. For full SO(3) rotations, we found there

were many box parameters that could represent the same box; for example, a box with $d = (1, 2, 3)$ is equivalent to a box with $d' = (2, 1, 3)$ and a 90° rotation. To account for this, we find all the box parameters $B = \{b^{(1)}, ..., b^{(m)}\}$ that represent the same box and supervise on all of them:

$$L(h, B) = -\frac{1}{|B|} \sum_{b^{(i)} \in B} \log(b^{(i)}|h) \tag{3}$$

4 Applying Autoregressive 3D Bounding Box Models

Given a trained autoregressive bounding-box model, how do we actually obtain predictions from it? There can be a few different options, depending on how the downstream application plans to use the predictions.

4.1 Beam Search

In many applications, we want to simply obtain the most likely 3D bounding box given the input observation. That is, we find the box $b^* = \arg\max_b p(b|h)$ which is most likely under the model. Finding b^* exactly can be computationally expensive, but we can approximate it using *beam search*, a technique that has proven especially popular for autoregressive models in natural language applications [3]. Beam search allows us to estimate the mode of the distribution learned by the model and serves as an effective pointwise prediction.

4.2 Quantile and Confidence Boxes

In applications such as robotics and autonomous driving, 3D bounding boxes are often used to estimate object extents and avoid collisions. To that end, we often care that an object o is fully contained in the estimated box b. For a given confidence requirement p, we define a confidence box b_p as a box that contains the true object o with probability at least p: $\mathbb{P}(o \subseteq b_p) \geq p$. We'll show how to use an autoregressive bounding box model for confidence box predictions.

Suppose we draw multiple samples K from our model. If a point $x \in \mathbb{R}^3$ is contained in many boxes, then it's likely that point is actually part of the object. Conversely, a point that is only contained in a few sampled boxes is not likely to be part of the object. We can formalize this intuition as the occupancy measure

$$O(x) = \mathbb{P}(x \in b) = \mathbb{E}_{b \sim p(b|h)}[\mathbb{1}\{x \in b\}] \approx \frac{1}{K} \sum_{i=1}^{K} \mathbb{1}\{x \in b^{(i)}\} \tag{4}$$

which can be approximated using samples $b^{(1)}, \ldots, b^{(K)} \sim p(b|h)$ from our model.

To find regions that are very likely to be part of an object, consider the set of all points that have occupancy greater than q:

$$Q(q) = \{x : O(x) > q\} \tag{5}$$

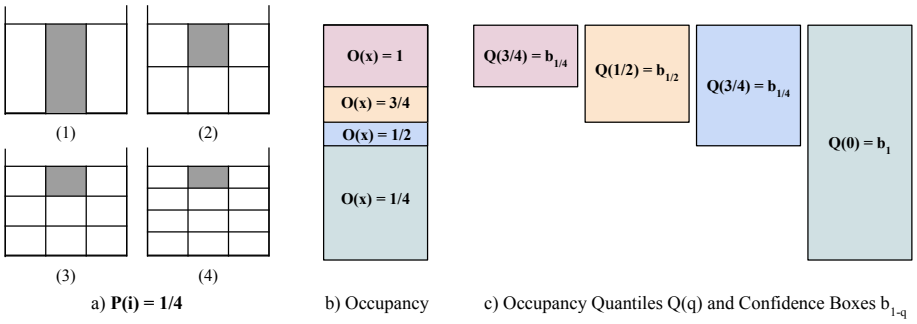

Fig. 4: Consider a scenario where we are estimating the bounding box of a tightly packed bin of stacked boxes. a) There is not enough visual information to estimate the object height, however, we know that the object could have heights H/i for $i \in \{1, 2, 3, 4\}$ with equal probability. b) We compute the occupancy $O(x)$ for different regions. c) We visualize occupancy quantiles $Q(q)$ which correspond to confidence boxes b_{1-q}. Notice that as the confidence requirement increases, the size of the box increases to ensure we can contain the true object.

which we'll refer to as the *occupancy quantile*. The minimum volume bounding box over the occupancy quantile is the *quantile box*:

$$b_q = \arg \min_{b: Q(q) \subseteq b} \text{vol}(b) \tag{6}$$

Under some conditions, we can show that quantile boxes are confidence boxes.

Theorem 1. *A quantile box b_q is a confidence box with confidence $p = 1 - q$ when $p(b|h)$ is an ordered object distribution.*

$p(b|h)$ is an ordered object distribution if for any two distinct boxes b_i, b_j in the sample space of $p(b|h)$, one box must be contained within the other, $b_i \subset b_j$ or $b_j \subset b_i$. Empirically we find that quantile boxes are good approximations for confidence boxes even when $p(b|h)$ is not an ordered object distribution. See Figure 4 for a visualization of occupancy and confidence boxes.

Quantile boxes provide an efficient way to make confidence box predictions with an autoregressive model. We can use the autoregressive distribution to estimate occupancy using supervision from 3D box labels (without requiring meshes for direct occupancy supervision). Occupancy quantiles provide a fast approach for confidence box estimation on ordered object distributions and a good confidence box approximation for general object distributions. Appendix B has the full proof of Theorem 1 and the details of our fast quantile box algorithm.

4.3 Uncertainty Measure

Uncertainty estimation is an important application of bounding box estimation. When the 3D extent of an object is unknown or not fully observed, it can be

valuable if a model can also indicate that its predictions are uncertain. For instance, a robot may choose to manipulate that uncertain object more slowly to avoid collisions, or an autonomous vehicle may be more cautious around a moving object of unknown size.

A pointwise predictor can accomplish this by predicting both a mean μ and variance σ^2 for each box parameter, maximizing a $\mathcal{N}(\mu, \sigma^2)$ likelihood [7]. However, the spread of the distribution is measured independently for each box parameter which doesn't measure the spread of the overall box distribution well.

With an autoregressive box parameterization, we can measure uncertainty in the space of boxes using quantile boxes. Let b_α and b_β be two quantile boxes with different quantiles. If we consider these boxes as confidence boxes, we can interpret (b_α, b_β) as a confidence interval or the spread of the box distribution. With this intuition, we can measure uncertainty using the IOU of different quantile boxes $U_{\alpha,\beta} = 1 - IOU(b_\alpha, b_\beta)$. This $U_{\alpha,\beta}$ effectively measures the span of the distribution in units of relative volume.

4.4 Dimension Conditioning

For some robotics applications, such as object manipulation in industrial settings, we are often presented with Stock-Keeping Unit, or SKU, information beforehand. In these scenarios, the dimensions of each SKU are provided, and the prediction task essentially boils down to correctly assigning the dimensions to a detected object instance, and predicting the pose of the 3D bounding box.

The autoregressive nature of our model allows for conveniently conditioning on the dimensions of each bounding box. However, we don't know which object in the scene corresponds to which SKU dimensions. How can we leverage dimension information from multiple SKUs without object-SKU correspondence? Our autoregressive model provides an elegant solution using conditioning and likelihood evaluation.

Given $\{d^{(1)}, ..., d^{(k)}\}$ known SKU dimensions, we can make a bounding box prediction using this information by maximizing:

$$b^* = \arg\max_b \{ \max_{d^{(1)},...,d^{(k)}} p(b|d^{(i)}, h) \} \tag{7}$$

We can find the optimal b^* by using beam search conditioned on each of the d_i and returning the box with the highest likelihood. Figure 1 shows an illustrative example of how dimension conditioning can be used to greatly increase the fidelity of the predicted 3D bounding boxes.

5 Experiments

We designed our experiments to answer the following questions:

1. How does an autoregressive bounding box predictor perform compared to a pointwise predictor, across a variety of domains and model architectures?
2. How meaningful are the uncertainty estimates from an autoregressive model? Are quantile boxes confidence boxes for general object distributions?

5.1 Datasets

To demonstrate the flexibility of our method, we conducted experiments on a diverse set of indoor, outdoor, and industrial datasets:

SUN-RGBD [23] is a real-world dataset containing monocular images and point clouds captured from a stereo depth camera. It features a large variety of indoor scenes and is one of the most popular benchmarks in 3D object detection. The box labels only include one rotational degree of freedom θ.

Scannet [2] is a dataset of indoor 3D reconstructions. There are 18 classes and box labels are axis-aligned (no rotation). We train on 1201 scenes and evaluate on 312 validation scenes.

KITTI [4] is a widely popular 3D detection dataset for autonomous driving. Objects in KITTI have one degree of rotational freedom θ, and we report evaluation results on the validation split.

COB-3D. *Common Objects in Bins 3D* is a simulated dataset rendered by Theory Studios to explore a qualitatively different set of challenges than the ones exhibited in popular datasets in the literature. We are releasing nearly 7000 scenes that aim to emulate industrial order-picking environments with each scene consisting of a bin containing a variety of items. There are two main themes we chose to highlight: first, the objects are in a greater range of orientations than any other 3D-bounding-box dataset. In particular, a model that performs well must reason about complete 3D rotations, whereas the state-of-the-art methods on SUN-RGBD only need to predict one rotational degree of freedom. Secondly, it exhibits many types of ambiguity including rotation symmetry, occlusion reasoning in cluttered scenes, and tightly-pack bins with unobserved dimensions. See Appendix C for full details on this dataset including visual examples.

5.2 Evaluation

To evaluate 3D-bounding-box predictions, *intersection-over-union*, or IoU, is commonly used to compare the similarity between two boxes. 3D object detection uses mean average precision, or mAP, to measure how well a detector trades off precision and recall. IoU is used to determine whether a prediction is close enough to a ground-truth box to constitute a true positive. For 3D bounding-box estimation, detection has already happened, so we simply measure the mean IoU between the prediction and ground-truth, averaged across objects.

Unlike 2D detection, many applications that use 3D bounding boxes especially care about underestimation more than overestimation: if the predicted bounding box is too large, that is generally a less costly error than if it is too small. In the latter case, there are parts of the object that are outside the bounding box, which may result in collisions in robotics or autonomous driving setting.

To help quantify this error asymmetry, we consider a new similarity functions, the *intersection-over-ground-truth* (IoG). IoG measure what fraction of the ground truth box is contained within the predicted box; when IoG is 1, the ground truth box is fully contained in the predicted box. With IoG and IoU, we have a more complete understanding of the types of errors that a bounding-box

Table 1: 3D Object Detection results on SUN-RGBD, Scannet, and KITTI

Dataset	Method	IoU			IoG		
		$AP_{0.25}$	$AP_{0.50}$	AP_{all}	$AP_{0.25}$	$AP_{0.50}$	AP_{all}
SUN-RGBD	FCAF3D	63.8	48.2	37.42	64.72	59.82	48.75
	3DETR	59.52	32.17	31.13	63.00	53.33	44.08
	VoteNet	60.71	38.98	30.25	62.81	54.58	43.62
	ImVoteNet	**64.24**	39.38	31.12	**67.00**	57.41	45.78
	Beam Search	62.94	47.03	38.15	64.75	58.50	47.17
	Quantile 0.1	61.21	30.94	31.06	65.89	**64.34**	**60.08**
	Quantile 0.4	63.46	48.41	38.43	65.34	61.68	51.76
	Quantile 0.45	63.47	**48.64**	**38.55**	65.19	61.03	50.36
	Quantile 0.5	63.30	47.70	38.50	64.99	59.83	48.44
Scannet	FCAF3D	68.53	**53.87**	43.32	72.05	67.63	60.66
	3DETR	64.09	47.16	39.57	68.62	59.17	49.82
	Beam Search	**69.06**	53.67	**43.85**	71.46	66.10	59.13
	Quantile 0.1	67.10	43.13	34.17	72.23	**70.01**	**66.73**
	Quantile 0.2	68.03	48.68	38.27	**72.30**	69.68	65.43
	Quantile 0.4	68.73	52.98	42.76	72.08	67.74	61.98
KITTI	Method	AP IoU Hard Split			AP IoG Hard Split		
		Car	Ped.	Cycl.	Car	Ped.	Cycl.
	PVRCNN	**82.37**	53.12	68.69	91.86	67.08	73.14
	Beam Search	**82.37**	52.28	**69.13**	91.84	66.96	73.40
	Quantile 0.1	59.75	39.26	58.38	**96.02**	**71.85**	**76.09**
	Quantile 0.4	81.98	**54.15**	68.45	93.98	70.63	74.08
	Quantile 0.5	82.32	53.78	69.03	91.84	68.14	73.52

predictor is making. For the detection task, we compute mAP separately using IoU and IoG, and for the estimation task, we compute the mean IoG along with the mean IoU.

5.3 3D Object Detection

To evaluate the autoregressive box parameterization for 3D Object Detection, we evaluate Autoregressive FCAF3D and Autoregressive PV-RCNN introduced in Section 3.2. Table 1 shows the comparison between autoregressive models and baselines on SUN-RGBD, Scannet, and KITTI. We find that beam search generally matches the baseline performance, if not exceeding performance on IoU AP_{all}.

As for quantile boxes, we find that lower quantiles result in higher IoG mAP which suggests that the predicted boxes are more likely to contain the ground truth box. This is consistent with our claim from Theorem 1 since lower quantiles correspond to higher confidence boxes and must contain the true object with higher probability. We find that quantile boxes 0.4-0.5 strike the best balance between IoU and IoG, achieving better mAP than baselines in most cases. This flexible quantile parameter enables applications to trade off bounding box

accuracy as measured by IoU with containment probability as measured by IoG. For instance, an autonomous vehicle may use a lower quantile to mitigate the risk of collisions at the cost of some bounding box accuracy.

5.4 3D Bounding Box Estimation

We evaluate the bounding box estimation on COB-3D using the model architecture described in Section 3.2. To compare the effectiveness of our autoregressive parameterization, we train the same model architecture with different box parameterizations and losses. All models receive the same 2D detection results and features as input and must make 3D bounding box predictions for each detected object. We consider 4 baseline parameterizations for this task inspired by various works in the literature:

L1 Regression: In this parameterization, the model outputs 9 real values for each of the 9 box parameters: $b = (d_x, d_y, d_z, c_x, c_y, c_z, \psi, \theta, \phi)$. The model predicts dimensions and centers in coordinates normalized around the object's point cloud. This model is trained using a L1 loss over the normalized box parameters $L(b, g) = ||b - g||_1$, where g is the ground truth box [13].

Gaussian: For this baseline, the model outputs 18 real values for the mean, μ, and log-variance, $\log \sigma^2$, of 9 Gaussian distributions $\mathcal{N}(\mu, \sigma)$ over the box parameters b [7,11]. Predicting the variance enables the model to output uncertainty over different box parameters, independently of each other. We train this model using maximum likelihood: $L(\mu, \log \sigma^2, g) = -\sum_i \log \mathcal{N}(g_i; \mu_i, \sigma_i)$.

Discrete: In some prior works, box parameters are predicted as discrete bins but not in an autoregressive manner [18]. To evaluate this parameterization and ablate the necessity of autoregressive predictions, we predict each box parameter *independently* as discrete bins: $\log p(b|h) = \sum_{i=1}^{9} \log p(b_i|h)$

4-Point: This baseline outputs 12 real values for four 3D corner points $(p_0, p_1, p_2, p_3) \in \mathbb{R}^3$, constituting a 3D bounding box [11,12]. We ensure that the 3D bounding box is orthogonal by applying the Gram-Schmidt process on the basis vectors $(p_1 - p_0, p_2 - p_0, p_3 - p_0)$. We use an L1-loss on the difference between the predicted points and the points of the ground truth 3D bounding box. Since there are many permutations of valid 4-point corners of a bounding box, we supervise on the permutation that induces the minimum loss.

Metrics. To make reasoning about the trade-off between IoG and IoU more quantifiable, we report the F1-score equivalent for this use case, i.e., $\text{F1}_{score} = \frac{2(IoU * IoG)}{IoU + IoG}$. We further report metrics on the dimension & pose errors, which are computed as follows:

- $err_{dim} = \text{sum}(|\mathbf{d} - \mathbf{d}_{gt}|)$, where we compute the error across all possible permutations and then choose the one with the smallest error.
- $err_{quat} = 2 \arccos(|\langle \mathbf{q}, \mathbf{q}_{gt} \rangle|)$, where $\mathbf{q}$ represents the rotational part of the pose as a quaternion. We compute the error across all possible symmetries and choose the one with the smallest error.
- $err_{center} = ||\mathbf{c} - \mathbf{c}_{gt}||_2$, where $\mathbf{c}$ is the 3D-center of the bounding box.

Table 2: Results of the proposed method & baselines on our dataset. We also show results for conditioning our method on ground truth dimensions

	IoU	IoG	F1	$err_{dim}[m]$	$err_{quat}[rad]$	$err_{center}[m]$
L1 Regression	0.4219	0.6113	0.4992	0.0436	0.4667	0.0138
Discrete	0.5232	0.6282	0.5709	0.0339	0.2926	**0.0105**
Gaussian	0.3169	0.5304	0.3967	0.0450	0.5154	0.0119
4-Point	0.5688	0.7113	0.6321	0.0332	0.1999	0.0132
Beam Search	**0.6296**	0.7877	0.6999	**0.0287**	**0.1598**	0.0109
Quantile 0.1	0.3821	**0.9723**	0.5486	0.0986	0.1762	0.0123
Quantile 0.4	0.5949	0.8871	0.7122	0.0377	0.1640	0.0110
Quantile 0.5	0.6275	0.8295	**0.7126**	0.0318	0.1657	0.0110
Conditioning	0.6709	0.7899	0.7215	0.0086	0.1674	0.0096

Results. Table 2 shows how our autoregressive methods compare to the baseline parameterizations. We find that *Beam Search* achieves the best IoU, dimension & rotation error. As for the *Quantile* methods, we find that lower quantiles achieve higher IoG while sacrificing IoU and dimension error. *Quantile 0.5* offers the best tradeoff in terms of overall performance, achieving higher IoG with similar IoU and dimension error compared to *Beam Search*. Baseline models that predict box parameters directly generally performed worse since those models cannot properly capture multimodal correlations across the box parameters. The *Discrete* baseline performs the best in terms of center error, but we can see that the best autoregressive methods are only a few millimeters worse. For bounding box predictions with full rotations in SO(3), we find that an autoregressive bounding box parameterization can effectively model rotation uncertainty, achieving the lowest rotation error. We can also see that conditioning the model on known dimensions of the items in the scene increases performance in all relevant metrics (besides IoG), most notably in IoU & dimension error. Note that the dimension error is non-zero because the model is given the dimensions as an unordered set, and still needs to predict the association of each dimension tuple to the corresponding item in the scene.

5.5 Quantile and Confidence Boxes

In Section 4.2 we introduced quantile boxes as a fast approximation for confidence boxes. We showed that when $p(b|h)$ is an ordered object distribution, a quantile box with quantile q is equivalent to a confidence box with $p = 1 - q$ and should contain the true object with probability p.

While it's hard to ensure that real world objects follow an ordered distribution, we can empirically evaluate whether q confidence boxes contain the ground truth object $1 - q$ fraction of the time. To test our hypothesis, we predict quantile boxes with different q and calculate the fraction of predictions f with IoG > 0.95. In Figure 5, we can see that $f \approx 1 - q$ and follows a generally linear

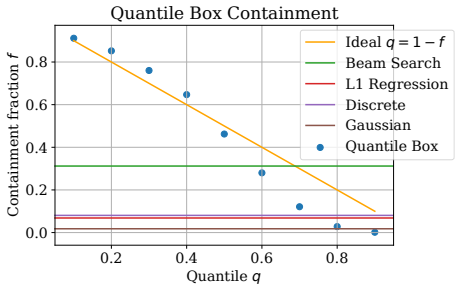

Fig. 5: We compare fraction of predicted boxes that contain ground truth boxes f with different quantiles q and find that q-quantile boxes contain approximately $f \approx 1-q$ fraction of ground truth boxes.

Method	ROC AUC	Spearman r_s
Gaussian	0.731	-0.530
Quantile 0.2	0.897	**-0.865**
Quantile 0.5	0.878	-0.789
Quantile 0.8	**0.967**	-0.850

Table 3: We compare Quantile Uncertainty Measure $U_{0.2,0.8}$ with Gaussian dimension variance G, and find that $U_{0.2,0.8}$ a better predictor of ground-truth IoU compared to G as measured by ROC AUC. $U_{0.2,0.8}$ is also better correlated with ground-truth IoU compared to G as measured by Spearman r_s

relationship. This suggests that even for general object distributions, quantile boxes can be an effective approximation for confidence boxes.

5.6 Uncertainty Measures

In Section 4.3, we introduced the uncertainty measure using quantile boxes $U_{\alpha,\beta} = IoU(b_\alpha, b_\beta)$ as a measure of the span of the confidence box interval. To evaluate the effectiveness of this uncertainty measure, we calculate the ROC AUC of using $U_{0.2,0.8}$ to predict when the IoU of the predicted box b with the ground truth box g is less than 0.25. We also measure the correlation between ground truth IoU and uncertainty using the Spearman's rank correlation r_s. We compare $U_{0.2,0.8}$ on different quantile boxes against Gaussian dimension variance $G = \frac{\sigma_{d_x}\sigma_{d_y}\sigma_{d_z}}{\mu_{d_x}\mu_{d_y}\mu_{d_z}}$ on the Gaussian baseline. Table 3 shows that quantile uncertainty $U_{0.2,0.8}$ can be a better uncertainty measure than G.

6 Discussion

We introduced an autoregressive formulation to 3D bounding prediction that greatly expands the ability of existing architectures to express uncertainty. We showed that it can be applied to both the 3D object detection and 3D bounding-box estimation settings, and explored different ways to extract bounding box predictions from such autoregressive models. In particular, we showed how the uncertainty expressed by these models can make high confidence predictions and meaningful uncertainty estimates. We introduced a dataset that requires predicting bounding boxes with full 3D rotations, and showed that our model naturally handles this task as well. While autoregressive models are just one class of distributionally expressive models, they are not the only option for more expressive bounding box modeling. We hope that future lines of work will continue to build upon the method, dataset, and benchmarks we introduced in this paper.

References

1. Choi, J., Chun, D., Kim, H., Lee, H.J.: Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 502–511 (2019)
2. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2017)
3. Freitag, M., Al-Onaizan, Y.: Beam search strategies for neural machine translation. In: Proceedings of the First Workshop on Neural Machine Translation. pp. 56–60. Association for Computational Linguistics, Vancouver (Aug 2017). https://doi.org/10.18653/v1/W17-3207, https://aclanthology.org/W17-3207
4. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) (2013)
5. Gilitschenski, I., Sahoo, R., Schwarting, W., Amini, A., Karaman, S., Rus, D.: Deep orientation uncertainty learning based on a bingham loss. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=ryloogSKDS
6. Hall, D., Dayoub, F., Skinner, J., Zhang, H., Miller, D., Corke, P., Carneiro, G., Angelova, A., Sünderhauf, N.: Probabilistic object detection: Definition and evaluation (11 2018)
7. He, Y., Zhu, C., Wang, J., Savvides, M., Zhang, X.: Bounding box regression with uncertainty for accurate object detection. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition. pp. 2888–2897 (2019)
8. Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J.: Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 21002–21012. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/f0bda020d2470f2e74990a07a607ebd9-Paper.pdf
9. Liu, Z., Zhang, Z., Cao, Y., Hu, H., Tong, X.: Group-free 3d object detection via transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2949–2958 (2021)
10. Metz, L., Ibarz, J., Jaitly, N., Davidson, J.: Discrete sequential prediction of continuous actions for deep rl. arXiv preprint arXiv:1705.05035 (2017)
11. Meyer, G.P., Laddha, A., Kee, E., Vallespi-Gonzalez, C., Wellington, C.K.: Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In: CVPR. pp. 12677–12686. Computer Vision Foundation / IEEE (2019), http://dblp.uni-trier.de/db/conf/cvpr/cvpr2019.html
12. Meyer, G.P., Thakurdesai, N.: Learning an uncertainty-aware object detector for autonomous driving. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 10521–10527 (2020)
13. Misra, I., Girdhar, R., Joulin, A.: An end-to-end transformer model for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2906–2917 (2021)
14. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3d bounding box estimation using deep learning and geometry. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7074–7082 (2017)

15. Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499 (2016)
16. Peretroukhin, V., Giamou, M., Rosen, D.M., Greene, W.N., Roy, N., Kelly, J.: A Smooth Representation of SO(3) for Deep Rotation Learning with Uncertainty. In: Proceedings of Robotics: Science and Systems (RSS'20) (Jul 12–16 2020)
17. Qi, C.R., Chen, X., Litany, O., Guibas, L.J.: Imvotenet: Boosting 3d object detection in point clouds with image votes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
18. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: ICCV. pp. 9276–9285. IEEE (2019), http://dblp.uni-trier.de/db/conf/iccv/iccv2019.html
19. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 918–927 (2018)
20. Rukhovich, D., Vorontsova, A., Konushin, A.: Fcaf3d: Fully convolutional anchor-free 3d object detection. arXiv preprint arXiv:2112.00322 (2021)
21. Shi, S., Wang, X., Li, H.P., et al.: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA. pp. 16–20 (2019)
22. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: PV-RCNN: point-voxel feature set abstraction for 3d object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 10526–10535. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.01054, https://openaccess.thecvf.com/content_CVPR_2020/html/Shi_PV-RCNN_Point-Voxel_Feature_Set_Abstraction_for_3D_Object_Detection_CVPR_2020_paper.html
23. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 567–576 (2015)
24. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (2019)
25. Van Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International conference on machine learning. pp. 1747–1756. PMLR (2016)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
27. Xie, Q., Lai, Y.K., Wu, J., Wang, Z., Zhang, Y., Xu, K., Wang, J.: Mlcvnet: Multi-level context votenet for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
28. Zhong, Y., Zhu, M., Peng, H.: Uncertainty-aware voxel based 3d object detection and tracking with von-mises loss. ArXiv **abs/2011.02553** (2020)
29. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5738–5746 (2019)