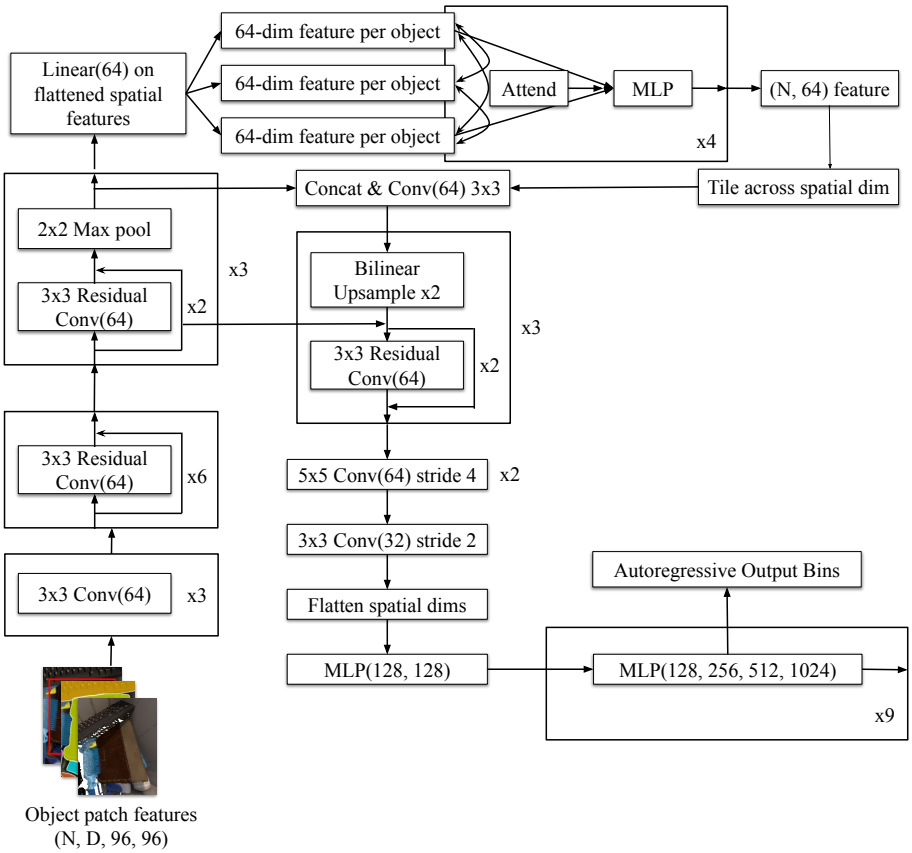

Fig. 6: Overview of Autoregressive Bounding Box Estimation architecture

A Model Architecture and Training

A.1 Autoregressive 3D Bounding Box Estimation

For bounding box estimation, our model operates on 2D detection patch outputs of size 96 x 96. We take the 2D bounding box from object-detection to crop and resize the following features for each object: 3D point cloud, depth uncertainty score, normals, instance mask, amodal instance mask (which includes the occluded regions of the object). We normalize each point p in the point cloud with the 0.25 (Q_1) and 0.75 (Q_3) quantiles per dimension using $\frac{p-c_0}{s}$ for $c_0 = \frac{Q_1+Q_3}{2}$, $s = Q_3 - Q_1$. We omitted RGB since we found it wasn't necessary for training and improved generalization.

We stack each 2D feature along the channel dimension and embed the features using a 2D Resnet U-Net. The features from the top of the U-Net are used in a

series of self-attention modules across embeddings from all objects in a scene so that information can be shared across objects. The resulting features from self-attention are tiled across the spatial dimension before the downward pass of the U-Net. Finally, the features from the highest spatial resolution of the U-Net are passed into several strided-convs, flattened, and projected to a 128-dimension feature h per object. Figure 6 shows the overview of our model architecture.

For the autoregressive layers, we use 9 MLPs with hidden layers (128, 256, 512, 1024). For baselines, we keep the same architecture through h and use different sized MLPs depending on the box parameterization. We train using Adam with learning rate 1e-5 with a batch size of 24 scenes per step with varying number of objects per scene. We train for 10000 steps or until convergence.

A.2 Autoregressive 3D Object Detection

For Autoregressive FCAF3D, we add 7 autoregressive MLPs with hidden dimensions (128, 256, 512). All other parameters of FCAF3D are the same and we train the same hyperparameters as the released code for 30 epochs. For the baseline FCAF3D, we trained the author-released model for 30 epochs on 8 gpus. We found that the benchmarked numbers for $AP_{0.25}$ and $AP_{0.50}$ were slightly lower than the reported ones in the original paper, so in our table, we use the reported average AP across trials from the original paper. AP_{all} was calculated in a similar way as in MS-COCO by averaging AP for iou thresholds over $0.05, 0.10, 0.15, ..., 0.95$.

B Quantile Box

B.1 Proof of Quantile-Confidence Box

Proof Sketch: Let $P(b)$ be a distribution over an ordered set of boxes where for any two distinct boxes b_1, b_2 in the sample space, one must be contained in the other, $b_1 \subset b_2$ or $b_2 \subset b_1$. We'll show that a quantile box b_q is a confidence box with $p = 1 - q$ by 1) constructing a confidence box b_p for any given q, 2) showing that any $x \in b_p$ must have $O(x) > q$, and 3) therefore $b_p \subseteq Q(q) \subseteq b_q$ so the quantile box is a confidence box.

1) Confidence Box: For any $p = 1-q$, we'll show how to construct a confidence box b_p. Using the ordered object distribution property of $P(b)$, we can define ordering as containment $b_1 < b_2 \equiv b_1 \subset b_2$. This ordering defines an inverse cdf:

$$F^{-1}(p) = \inf\{x : P(b \leq x) \geq p\} \tag{8}$$

Let $b_p = F^{-1}(1 - q)$ be the inverse cdf of p; by definition b_p is a confidence box with confidence p since $P(b \leq b_p) = P(b \subseteq b_p) \geq p$

2) Occupancy of b_p: We'll show that any $x \in b_p$ satisfies $O(x) > 1 - p$. First we'll prove that that $P(b \geq b_p) > 1-p$. Let $b_0 = \inf\{b : b < b_p\}$, the smallest box that is strictly contained in b_p. (If no such b_0 exists, then b_p must be the smallest box in the distribution order such that $P(b \geq b_p) = 1$ and $P(b \geq b_p) > 1 - p$ for $p \neq 0$)

Since b_p is the inverse cdf of p, we know that $P(b \leq b_0) < p$, otherwise b_0 would be the inverse cdf of p (i.e. $b_0 = b_p$ a contradiction). It follows that

$$\begin{aligned} P(b \geq b_p) &= P(b > b_0) && (9)\\ &= 1 - P(b \leq b_0) && (10)\\ &> 1 - p && (11) \end{aligned}$$

Now consider any point $x \in b_p$:

$$\begin{aligned} O(x) &= P(x \in b) && (12)\\ &= \int_b \mathbb{1}\{x \in b\} p(b) db && (13)\\ &\geq \int_{b \geq b_p} \mathbb{1}\{x \in b\} p(b) db && (14)\\ &= \int_{b \geq b_p} p(b) db && (15)\\ &= P(b \geq b_p) && (16)\\ &> 1 - p && (17) \end{aligned}$$

Where (14) follows from the nonegativity of $\mathbb{1}\{x \in b\}p(b)$. (15) follows from $x \in b_p$, $b_p \subseteq b$ which implies $x \in b$.

3) Quantile-Confidence Box: Since any $x \in b_p$ satisfies $O(x) > 1 - p$, it follows that $b_p \subseteq Q(1 - p)$, where $Q(q) = \{x : O(x) > q\}$ is the occupancy quantile with quantile q. The quantile box by construction must contain the occupancy quantile $Q(q) \subseteq b_q$, therefore we have $b_p \subseteq Q(1 - p) \subseteq b_q$, and

$$\begin{aligned} P(b \subseteq b_q) &\geq P(b \subseteq b_p) && (18)\\ &\geq p && (19) \end{aligned}$$

So b_q is a confidence box with confidence requirement p.

B.2 Quantile Box Algorithm

We propose a fast quantile box Algorithm 1 that runs in polynomial time and is easily batchable on GPU. We use a finite sample of k boxes to approximate the occupancy and a sample of km points to approximate the occupancy quantile $Q(q)$. To find the minimum volume box, we assume that one of the sampled box rotations will be close to the optimal quantile box rotation. We take the sampled

Algorithm 1: Quantile Box Algorithm

Given: quantile q, box distribution $P(b|h)$, numbers of box samples k, number of point samples m
Sample $b^{(1)}, ..., b^{(k)} \sim P(b|h)$ boxes
For each $b^{(i)}$, sample m random points within $b^{(i)}$, adding all points to a set T
For all $x \in T$, estimate $O(x) = \frac{1}{k} \sum_i^k \mathbb{1}\{x \in b^{(i)}\}$
Construct the occupancy quantile $Q(q) = \{x \in T : O(x) > q\}$
for $b^{(1)}, ..., b^{(k)}$ **do**
  Let R_i be the rotation of $b^{(i)}$
  Compute the volume of the $Q(q)$ bounding box under R_i, $v_i = \prod_{a \in x,y,z} (\max_{x \in Q(q)} (R_i^{-1} x)_a - \min_{x \in Q(q)} (R_i^{-1} x)_a)$
Find the minimum volume box $i^* = \arg\min_i v_i$
Let $s_a = \max_{x \in Q(q)} (R_{i^*}^{-1} x)_a$, $t_a = \min_{x \in Q(q)} (R_{i^*}^{-1} x)_a$
Return box $b = (d, c, R_{i^*})$ with dimensions $d = (t_x - s_x, t_y - s_y, t_z - s_z)$ and center $c = R_{i^*}(s_x + d_x/2, s_y + d_y/2, s_z + d_z/2)$

rotations and calculate the rotation-axis-aligned bounding box volume for the occupancy quantile. The minimum volume rotation is selected for the quantile box and corresponding dimension/center calculated accordingly.

Empirically we find that $k = 64$, $m = 4^3$ provides a good trade-off of variance and inference time. We can efficiently batch all operations on GPU, and find that quantile box inference for 15 objects takes no more than 10ms on a NVIDIA 1080TI.

C Dataset

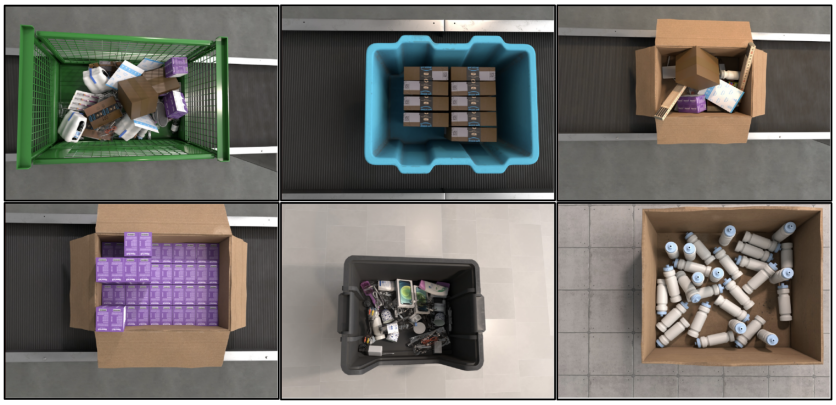

Fig. 7: Examples of scenes from our dataset

Our dataset consists of almost 7000 simulated scenes of common objects in bins. See Figure 7 for examples. Each scene consists of the following data:

- **RGB** image of shape (H, W, 3)
- **Depth** map of shape (H, W)
- **Intrinsic Matrix** of the camera (3, 3)
- **Normals Map** of shape (H, W, 3)
- **Instance Masks** of shape (N, H, W) where N is the number of objects
- **Amodal Instance masks** of shape (N, H, W) which includes the occluded regions of the object
- **3D Bounding Box** of each object (N, 9) as determined by dimensions, center, and rotation.

D Visualizations

In this section, we show various qualitative comparisons and visualization of our method.

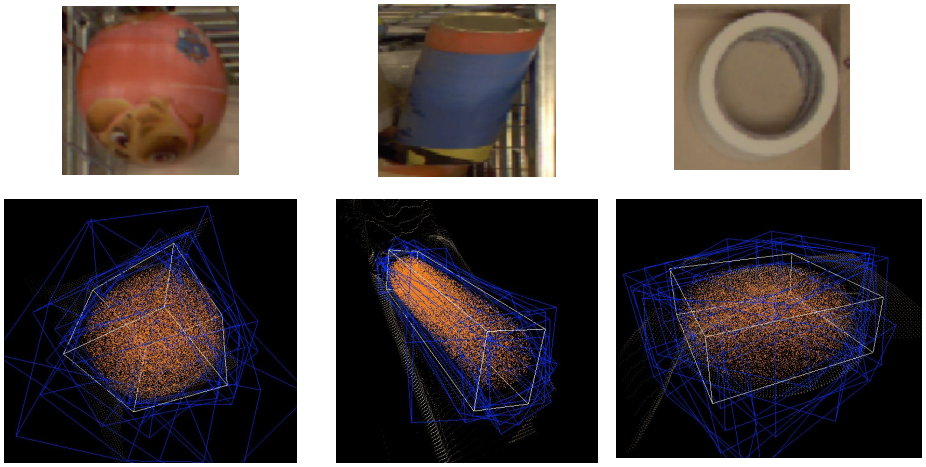

Fig. 8: Visualization of our model predictions on objects with rotational symmetry. The blue boxes show various samples from our model. The orange point cloud is the occupancy quantile. The white box is the quantile box.

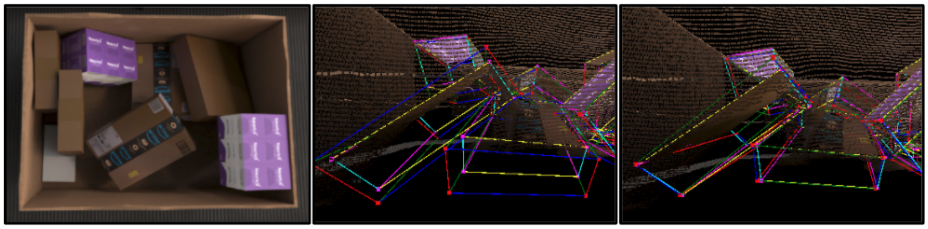

Fig. 9: Visualization of our dimension conditioning method. The model is able to leverage the conditioning information to accurately predict the correct pose & dimension for each object's 3D bounding box. The prediction is shown in red-blue-green and the ground truth in turquoise-yellow-pink. Left: image of the scene. Middle: vanilla beam search. Right: beam search with dimension conditioning.

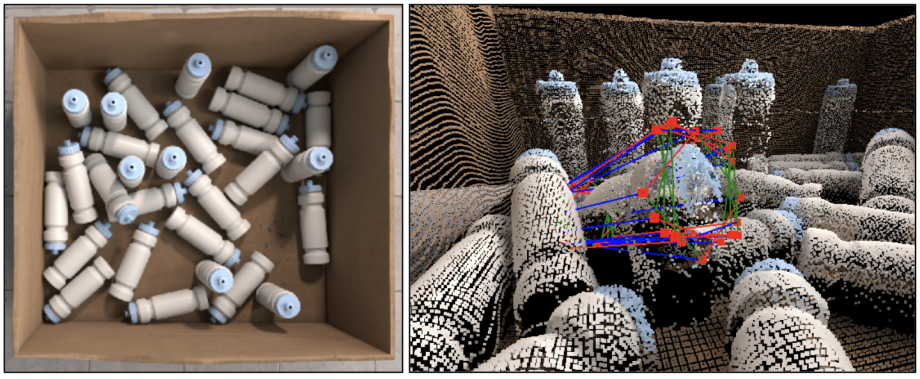

Fig. 10: Visualization of bounding box samples from our autoregressive model on a rotationally symmetric water bottle. Our model is able to sample different modes for symmetric objects whereas a deterministic model would only be able to predict a single mode.

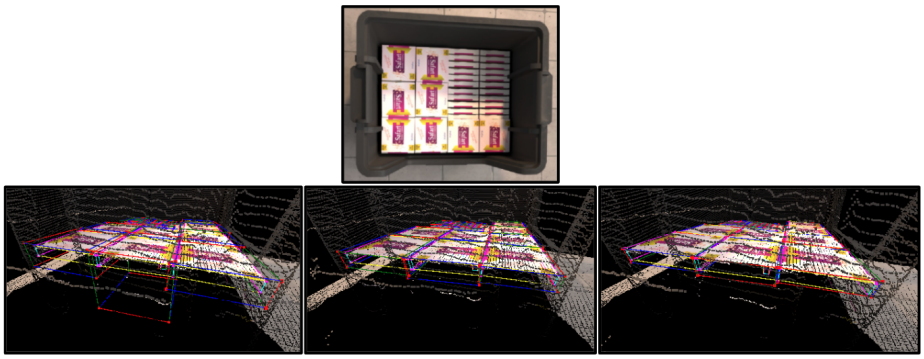

Fig. 11: Visualization of bounding box predictions with different quantiles. We can see that lower quantiles lead to larger boxes in the direction of uncertainty. Top: image of the scene. Left: quantile 0.1 Middle: quantile 0.3. Right: quantile 0.5.