# LidarNAS: Unifying and Searching Neural Architectures for 3D Point Clouds (Supplementary Material)

Chenxi Liu, Zhaoqi Leng, Pei Sun, Shuyang Cheng, Charles R. Qi, Yin Zhou, Mingxing Tan, and Dragomir Anguelov

Waymo LLC
{cxliu, lengzhaoqi, peis, shuyangcheng, rqi, yinzhou, tanmingxing, dragomir}@waymo.com

## A Anchor-Free Detection Head

We describe the details of our anchor-free detection head that works across views and formats. The key is to abstract away from the specific views and formats, and think about the individual *elements*. The elements are individual voxels / pixels / pillars under the voxel / perspective / pillar view.

### A.1 Training Phase

The detection head has two sequential jobs: finding the centers, and regressing parameters from them.

**Finding the Centers** In training the network to find the centers, we construct a ground truth *heatmap*. For each element $e \in E$ where $E$ is the set of all elements, we use $V(e)$ to represent its Cartesian coordinates, which can be either 2-dimensional (only $x$ and $y$) or 3-dimensional (all of $x, y, z$). We construct its ground truth heatmap value to be:

$$h(e) = \max_{c \in C(e)} \exp(-\frac{||V(e) - c|| - \min_{f \in E} ||V(f) - c||}{\sigma^2})$$

where $C(e)$ is the set of centers of the boxes that contain $e$, and $\sigma$ is a hyper-parameter. $h(e) = 0$ if $|C(e)| = 0$. Intuitively, the heatmap value is high when the element is close to an object center ($||V(e) - c||$). This distance is modified / compensated by the closest distance among all the elements ($\min_{f \in E} ||V(f) - c||$).

A penalty-reduced focal loss [1, 5] is used to train the predicted heatmap:

$$L_{\text{center}} = -\frac{1}{|E|} \sum_{e \in E} \{(1 - \tilde{h}(e))^{\alpha} \log(\tilde{h}(e)) \mathbb{I}_{h(e) > 1-\epsilon} + (1 - h(e))^{\beta} \tilde{h}(e)^{\alpha} \log(1 - \tilde{h}(e)) \mathbb{I}_{h(e) \leq 1-\epsilon}\}$$

where $\tilde{h}(e)$ is the predicted heatmap value for element $e$, $\alpha = 2$, $\beta = 4$, $\epsilon = 0.001$.

**Regressing Box Parameters** We use smooth L1 loss to regress the 3-dimensional box center offsets, as well as the 3-dimensional box length, width, height. We use a bin loss [2] to regress the heading. We also add a IoU loss [4]. These losses are only active for elements that have ground truth heatmap values greater than a threshold $\delta$.

### A.2 Inference Phase

After the forward pass produces the predicted heatmap $\tilde{h}$, the predicted object centers are the elements whose predicted heatmap value exceeds a threshold *and* is the local maximum. The latter is achieved by max pooling (possible on both dense grids and sparse) within a local window ($3 \times 3$ or $3 \times 3 \times 3$). The box parameters prediction on these elements complete the inference.

We conclude by reiterating that when the view is voxel and the format is sparse, this detection head exactly follows RSN [3].

## B Randomly Generated Architectures

We describe our procedure of randomly generating architectures stage by stage.

For the first stage, we add each view with probability 0.5 independently. For views that may have either dense or sparse formats, the format is selected with equal probability. The pillar / voxel size is 0.32m to avoid voxelization mismatch complications. The number of channels is 32 multiplied by either 0.8 or 1.0 or 1.2. The layer progression is randomly chosen between five choices. If the layer type is point, this means the number of dense-normalization-ReLU is between 1 and 5. For the other layer types, this means the number of downsampling / upsampling scales choose between $(0, 0)$, $(1, 0)$, $(2, 0)$, $(2, 1)$, $(2, 2)$.

For the second stage, we again add each view with probability 0.5 independently. For each added view, we iterate through the views selected in the first stage, and add it to the ancestor with probability 0.5 independently. The generation process for the other parameters (pillar / voxel size, number of channels, layer progression) is the same as the first stage.

The third stage should only contain one view. We select the view among voxel, perspective, pillar with equal probability. For views that may have either dense or sparse formats, the format is selected with equal probability. For this selected view, all the views selected in the second stage are its ancestors. The generation process for the other parameters is the same as the preceding stages.

The randomly generated architecture may be invalid for several reasons. Examples include: no branches are added in a particular stage; no ancestors are selected for a second stage view; there may be views in the first stage that are not selected by any view in the second stage. If any of these situations happen, we reject the sample and sample again until it succeeds.

## References

1. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
2. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 770–779 (2019)
3. Sun, P., Wang, W., Chai, Y., Elsayed, G., Bewley, A., Zhang, X., Sminchisescu, C., Anguelov, D.: Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5725–5734 (2021)
4. Zhou, D., Fang, J., Song, X., Guan, C., Yin, J., Dai, Y., Yang, R.: Iou loss for 2d/3d object detection. In: 2019 International Conference on 3D Vision (3DV). pp. 85–94. IEEE (2019)
5. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)