GrooMeD-NMS: Grouped Mathematically Differentiable NMS for Monocular 3D Object Detection

Abhinav Kumar, Garrick Brazil, Xiaoming Liu
Michigan State University, East Lansing, MI, USA
[kumarab6, brazilga, liuxm]@msu.edu
https://github.com/abhi1kumar/groomed_nms

Abstract

Modern 3D object detectors have immensely benefited from the end-to-end learning idea. However, most of them use a post-processing algorithm called Non-Maximal Suppression (NMS) only during inference. While there were attempts to include NMS in the training pipeline for tasks such as 2D object detection, they have been less widely adopted due to a non-mathematical expression of the NMS. In this paper, we present and integrate GrooMeD-NMS – a novel Grouped Mathematically Differentiable NMS for monocular 3D object detection, such that the network is trained end-to-end with a loss on the boxes after NMS. We first formulate NMS as a matrix operation and then group and mask the boxes in an unsupervised manner to obtain a simple closed-form expression of the NMS. GrooMeD-NMS addresses the mismatch between training and inference pipelines and, therefore, forces the network to select the best 3D box in a differentiable manner. As a result, GrooMeD-NMS achieves state-of-the-art monocular 3D object detection results on the KITTI benchmark dataset performing comparably to monocular video-based methods.

1. Introduction

3D object detection is one of the fundamental problems in computer vision, where the task is to infer 3D information of the object. Its applications include augmented reality [2, 68], robotics [43, 74], medical surgery [70], and, more recently path planning and scene understanding in autonomous driving [17, 35, 46, 77]. Most of the 3D object detectors [17, 35, 44, 46, 77] are extensions of the 2D object detector Faster R-CNN [69], which relies on the end-to-end learning idea to achieve State-of-the-Art (SoTA) object detection. Some of these methods have proposed changing architectures [46, 76, 77] or losses [10, 18]. Others have tried incorporating confidence [12, 76, 77] or temporal cues [12].

Almost all of them output a massive number of boxes for each object and, thus, rely on post-processing with a greedy [65] clustering algorithm called Non-Maximal Suppression (NMS) during inference to reduce the number of false positives and increase performance. However, these works have largely overlooked NMS’s inclusion in training leading to an apparent mismatch between training and inference pipelines as the losses are applied on all boxes before NMS but not on final boxes after NMS (see Fig. 1(a)).
We also find that 3D object detection suffers a greater mismatch between classification and 3D localization compared to that of 2D localization, as discussed further in Sec. A3.2 of the supplementary and observed in [12, 35, 76]. Hence, our focus is 3D object detection.

Earlier attempts to include NMS in the training pipeline [31, 32, 65] have been made for 2D object detection where the improvements are less visible. Recent efforts to improve the correlation in 3D object detection involve calculating [77, 79] or predicting [12, 76] the scores via likelihood estimation [40] or enforcing the correlation explicitly [35]. Although this improves the 3D detection performance, improvements are limited as their training pipeline is not end to end in the absence of a differentiable NMS.

To address the mismatch between training and inference pipelines as well as the mismatch between classification and 3D localization, we propose including the NMS in the training pipeline, which gives a useful gradient to the network so that it figures out which boxes are the best-localized in 3D and, therefore, should be ranked higher (see Fig. 1(b)).

An ideal NMS for inclusion in the training pipeline should be not only differentiable but also parallelizable. Unfortunately, the inference-based classical NMS and SoftNMS [8] are greedy, set-based and, therefore, not parallelizable [65]. To make the NMS parallelizable, we first formulate the classical NMS as matrix operation and then obtain a closed-form mathematical expression using elementary matrix operations such as matrix multiplication, matrix inversion, and clipping. We then replace the threshold pruning in the classical NMS with its softer version [8] to get useful gradients. These two changes make the NMS GPU-friendly, and the gradients are backpropagated. We next group and mask the boxes in an unsupervised manner, which removes the matrix inversion and simplifies our proposed differentiable NMS expression further. We call this NMS as Grouped Mathematically Differentiable Non-Maximal Suppression (GrooMeD-NMS).

In summary, the main contributions of this work include:

- This is the first work to propose and integrate a closed-form mathematically differentiable NMS for object detection, such that the network is trained end-to-end with a loss on the boxes after NMS.
- We propose an unsupervised grouping and masking on the boxes to remove the matrix inversion in the closed-form NMS expression.
- We achieve SoTA monocular 3D object detection performance on the KITTI dataset performing comparably to monocular video-based methods.

2. Related Work

**3D Object Detection.** Recent success in 2D object detection [26, 27, 48, 67, 69] has inspired people to infer 3D information from a single 2D (monocular) image. However, the monocular problem is ill-posed due to the inherent scale/depth ambiguity [82]. Hence, approaches use additional sensors such as LiDAR [35, 75, 88], stereo [45, 87] or radar [58, 84]. Although LiDAR depth estimations are accurate, LiDAR data is sparse [33] and computationally expensive to process [82]. Moreover, LiDARs are expensive and do not work well in severe weather [82].

Hence, there have been several works on monocular 3D object detection. Earlier approaches [15, 23, 61, 62] use hand-crafted features, while the recent ones are all based on deep learning. Some of these methods have proposed changing architectures [46, 49, 82] or losses [10, 18]. Others have tried incorporating confidence [12, 49, 76, 77], augmentation [80], depth in convolution [10, 22] or temporal cues [12]. Our work proposes to incorporate NMS in the training pipeline of monocular 3D object detection.

**Non-Maximal Suppression.** NMS has been used to reduce false positives in edge detection [72], feature point detection [29, 53, 57], face detection [85], human detection [11, 13, 20] as well as SoTA 2D [26, 48, 67, 69] and 3D detection [4, 12, 17, 76, 77, 82]. Modifications to NMS in 2D detection [8, 21, 31, 32, 65], 2D pedestrian detection [42, 51, 73], 2D salient object detection [91] and 3D detection [76] can be classified into three categories – inference NMS [8, 76], optimization-based NMS [3, 21, 42, 73, 86, 91] and neural network based NMS [30–32, 51, 65].

The inference NMS [8] changes the way the boxes are pruned in the final set of predictions. [76] uses weighted averaging to update the z-coordinate after NMS. [73] solves quadratic unconstrained binary optimization while [3, 42, 81] and [91] use point processes and MAP based inference respectively. [21] and [86] formulate NMS as a structured prediction task for isolated and all object instances respectively. The neural network NMS use a multi-layer network and message-passing to approximate NMS [31, 32, 65] or to predict the NMS threshold adaptively [51]. [30] approximates the sub-gradients of the network without modelling NMS via a transitive relationship. Our work proposes a grouped closed-form mathematical approximation of the classical NMS and does not require multiple layers or message-passing. We detail these differences in Sec. 4.2.

3. Background

3.1. Notations

Let \( B = \{b_i\}_{i=1}^{n} \) denote the set of boxes or proposals \( b_i \) from an image. Let \( s = \{s_i\}_{i=1}^{n} \) and \( r = \{r_i\}_{i=1}^{n} \) denote their scores (before NMS) and rescores (updated scores after NMS) respectively such that \( r_i, s_i \geq 0 \; \forall \; i \). \( D \) denotes the subset of \( B \) after the NMS. Let \( O = [o_{ij}] \) denote the \( n \times n \) matrix with \( a_{ij} \) denoting the 2D Intersection over Union (IoU_{2D}) of \( b_i \) and \( b_j \). The pruning function \( p \) decides how to rescore a set of boxes \( B \) based on IoU_{2D} overlaps.
of its neighbors, sometimes suppressing boxes entirely. In other words, \( p(o_i) = 1 \) denotes the box \( b_i \) is suppressed while \( p(o_i) = 0 \) denotes \( b_i \) is kept in \( D \). The NMS threshold \( N_i \) is the threshold for which two boxes need in order for the non-maximum to be suppressed. The temperature \( \tau \) controls the shape of the exponential and sigmoidal pruning functions \( p, v \) thresholds the rescores in GrooMeD and Soft-NMS [9] to decide if the box remains valid after NMS.

\( B \) is partitioned into different groups \( G = \{G_k\} \). \( B_{G_k} \) denotes the subset of \( B \) belonging to group \( k \). Thus, \( B_{G_k} = \{b_i \in G_k \} \) and \( B_{G_k} \cap B_{G_l} = \emptyset \) for \( k \neq l \). \( G_k \) in the subscript of a variable denotes its subset corresponding to \( B_{G_k} \). Thus, \( s_{G_k} \) and \( r_{G_k} \) denote the scores and the rescores of \( B_{G_k} \) respectively. \( \alpha \) denotes the maximum group size.

\( \lor \) denotes the logical OR while \( \lfloor x \rfloor \) denotes clipping of \( x \) in the range \([0, 1]\). Formally,

\[
\lfloor x \rfloor = \begin{cases} 
1, & x > 1 \\
0, & x < 0 \\
0, & 0 \leq x \leq 1 
\end{cases}
\]

\(|s|\) denotes the number of elements in \( s \). \( \mathbb{I}_n \) in the subscript denotes the lower triangular version of the matrix without the principal diagonal. \( \odot \) denotes the element-wise multiplication. \( \mathbb{I} \) denotes the identity matrix.

3.2. Classical and Soft-NMS

NMS is one of the building blocks in object detection whose high-level goal is to iteratively suppress boxes which have too much IoU with a nearby high-scoring box. We first give an overview of the classical and Soft-NMS [8], which are greedy and used in inference. Classical NMS uses the idea that the score of a box having a high IoU2D overlap with any of the selected boxes should be suppressed to zero. That is, it uses a hard pruning \( p \) without any temperature \( \tau \). Soft-NMS makes this pruning soft via temperature \( \tau \). Thus, classical and Soft-NMS only differ in the choice of \( p \). We reproduce them in Alg. 1 using our notations.

4. GrooMeD-NMS

Classical NMS (Alg. 1) uses argmax and greedily calculates the rescore \( r_i \) of boxes \( B \) and, is thus not parallelizable or differentiable [65]. We wish to find its smooth approximation in closed-form for including in the training pipeline.

4.1. Formulation

4.1.1 Sorting

Classical NMS uses the non-differentiable hard argmax operation (Line 6 of Alg. 1). We remove the argmax by hard sorting the scores \( s \) and \( O \) in decreasing order (lines 2-3 of Alg. 2). We also try making the sorting soft. Note that we require the permutation of \( s \) to sort \( O \). Most soft sorting
methods [6, 7, 60, 63] apply the soft permutation to the same vector. Only two other methods [19, 64] can apply the soft permutation to another vector. Both methods use $O(n^2)$ computations for soft sorting [7]. We implement [64] and find that [64] is overly dependent on temperature $\tau$ to break out the ranks, and its gradients are too unreliable to train our model. Hence, we stick with the hard sorting of $s$ and $O$.

### 4.1.2 NMS as a Matrix Operation

The rescoring process of the classical NMS is greedy set-oriented, and its gradients are too unreliable to train our model. However, to avoid recursion, we use the rescores $r$ in a matrix formulation as

$$r_i \approx \max \left( s_i - \sum_{j=1}^{i-1} p(a_{ij})r_j, 0 \right)$$

(2)

using the relaxation of logical OR $\lor$ operator as $\sum [38, 47]$. See Sec. A1 of the supplementary material for an alternate explanation of (2). The presence of $r_j$ on the RHS of (2) prevents suppressed boxes from influencing other boxes hugely. When $p$ outputs discretely as $\{0, 1\}$ as in classical NMS, scores $s_i$ are guaranteed to be suppressed to $r_i = 0$ or left unchanged $r_i = s_i$ thereby implying $r_i \leq s_i \forall i$. We write the rescores $r$ in a matrix formulation as

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \approx \max \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix},$$

(3)

with

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} - \begin{bmatrix} 0 & 0 & \ldots & 0 \\ p(o_{21}) & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p(o_{n1}) & p(o_{n2}) & \ldots & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}.$$  

(4)

The above two equations are written compactly as

$$r \approx \max(s - Pr, 0),$$

(5)

where $P$, called the Prune Matrix, is obtained when the pruning function $p$ operates element-wise on $O_L$. Maximum operation makes (5) non-linear [41] and, thus, difficult to solve. However, to avoid recursion, we use

$$r \approx \begin{bmatrix} I + P \end{bmatrix}^{-1} s,$$  

(6)

and the boxes have overlaps, the rescores are the scores divided by a number greater than one and are, therefore, lesser than scores. If the boxes do not overlap, the division is by one and rescores equal scores.

Note that the $I + P$ in (6) is a lower triangular matrix with ones on the principal diagonal. Hence, $I + P$ is always full rank and, therefore, always invertible.

### 4.1.3 Grouping

We next observe that the object detectors output multiple boxes for an object, and a good detector outputs boxes wherever it finds objects in the monocular image. Thus, we cluster the boxes in an image in an unsupervised manner based on IoU$_{2D}$ overlaps to obtain the groups $G$. Grouping thus mimics the grouping of the classical NMS, but does not rescore the boxes. As clustering limits interactions to intra-group interactions among the boxes, we write (6) as

$$r_{G_k} \approx \left( I_{G_k} + P_{G_k} \right)^{-1} s_{G_k}.$$  

(7)

This results in taking smaller matrix inverses in (7) than (6).

We use a simplistic grouping algorithm, i.e., we form a group $G_k$ with boxes having high IoU$_{2D}$ overlap with the top-ranked box, given that we sorted the scores. As the group size is limited by $\alpha$, we choose a minimum of $\alpha$ and the number of boxes in $G_k$. We next delete all the boxes of this group and iterate until we run out of boxes. Also, grouping uses IoU$_{2D}$ since we can achieve meaningful clustering in 2D. We detail this unsupervised grouping in Alg. 3.

### 4.1.4 Masking

Classical NMS considers the IoU$_{2D}$ of the top-scored box with other boxes. This consideration is equivalent to only keeping the column of $O$ corresponding to the top box while assigning the rest of the columns to be zero. We implement this through masking of $P_{G_k}$. Let $M_{G_k}$ denote the binary mask corresponding to group $G_k$. Then, entries in the binary matrix $M_{G_k}$ in the column corresponding to the top-scored box are 1 and the rest are 0. Hence, only one of the columns in $M_{G_k} \odot P_{G_k}$ is non-zero. Now, $I_{G_k} + M_{G_k} \odot P_{G_k}$ is a Frobenius matrix (Gaussian transformation) and we, therefore, invert this matrix by simply subtracting the second term [28]. In other words, $(I_{G_k} + M_{G_k} \odot P_{G_k})^{-1} = I_{G_k} - M_{G_k} \odot P_{G_k}$. Hence, we simplify (7) further to get

$$r_{G_k} \approx \left[ I_{G_k} - M_{G_k} \odot P_{G_k} \right] s_{G_k}.$$  

(8)

Thus, masking allows to bypass the computationally expensive matrix inverse operation altogether.

We call the NMS based on (8) as Grouped Mathematically Differentiable Non-Maximal Suppression or GrooMeD-NMS. We summarize the complete GrooMeD-NMS in Alg. 2 and show its block-diagram in Fig. 1(c).
GrooMeD-NMS in Fig. 1(c) provides two gradients - one through \( s \) and other through \( O \).

### 4.1.5 Pruning Function

As explained in Sec. 3.1, the pruning function \( p \) decides whether to keep the box in the final set of predictions \( D \) or not based on IOU \( D \) overlaps, i.e., \( p(o_i) = 1 \) denotes the box \( b_i \) is suppressed while \( p(o_i) = 0 \) denotes \( b_i \) is kept in \( D \).

Classical NMS uses the threshold as the pruning function, which does not give useful gradients. Therefore, we considered three different functions for \( p \): Linear, a temperature \( (\tau) \)-controlled Exponential, and Sigmoidal function.

- **Linear** Linear pruning function \([8]\) is \( p(o) = o \).
- **Exponential** Exponential pruning function \([8]\) is \( p(o) = 1 - \exp \left( -\frac{\tau}{\tau} \right) \).
- **Sigmoidal** Sigmoidal pruning function is \( p(o) = \sigma \left( \frac{2 \cdot o - 1}{\tau} \right) \) with \( \sigma \) denoting the standard sigmoid. Sigmoidal function appears as the binary cross entropy relaxation of the subset selection problem \([60]\).

We show these pruning functions in Fig. 2. The ablation studies (Sec. 5.4) show that choosing \( p \) as Linear yields the simplest and the best GrooMeD-NMS.

### 4.2 Differences from Existing NMS

Although no differentiable NMS has been proposed for the monocular 3D object detection, we compare our GrooMeD-NMS with the NMS proposed for 2D object detection, 2D pedestrian detection, 2D salient object detection, and 3D object detection in Tab. 1. No method described in Tab. 1 has a matrix-based closed-form mathematical expression of the NMS. Classical, Soft \([8]\) and Distance-NMS \([76]\) are used at the inference time, while GrooMeD-NMS is used during both training and inference. Distance-NMS \([76]\) updates the \( z \)-coordinate of the box after NMS as the weighted average of the \( z \)-coordinates of top-\( K \) boxes. QUBO-NMS \([73]\), Point-NMS \([42,81]\), and MAP-NMS \([91]\) are not used in end-to-end training. \([3]\) proposes a trainable Point-NMS. The Structured-SVM based NMS \([21,86]\) rely on structured SVM to obtain the rescores.

Adaptive-NMS \([51]\) uses a separate neural network to predict the classical NMS threshold \( N_l \). The trainable neural network based NMS (NN-NMS) \([31,32,65]\) use a separate neural network containing multiple layers and/or message-passing to approximate the NMS and do not use the pruning function. Unlike these methods, GrooMeD-NMS uses a single layer and does not require multiple layers or message passing. Our NMS is parallel up to group (denoted by \( \mathcal{G} \)). However, \( \mathcal{G} \) is, in general, \( \ll \) \( \mathcal{B} \) in the NMS.

### 4.3 Target Assignment and Loss Function

#### Target Assignment.

Our method consists of M3D-RPN \([10]\) and uses binning and self-balancing confidence \([12]\). The boxes’ self-balancing confidence are used as scores \( s \), which pass through the GrooMeD-NMS layer to obtain the rescores \( r \). The rescores signal the network if the best box has not been selected for a particular object.

We extend the notion of the best 2D box \([65]\) to 3D. The best box has the highest product of IOU \( D \) and gIoU \( D \) with ground truth \( g_i \). If the product is greater than a certain threshold \( \beta \), it is assigned a positive label. Mathematically,

\[
\text{target}(b_i) = \begin{cases} 
1, & \text{if } \exists g_i \text{ st } i = \text{argmax } q(b_j, g_i) \\
0, & \text{otherwise}
\end{cases}
\]

with \( q(b_j, g_i) = \text{IOU}_{2D}(b_j, g_i) \left( \frac{1 + \text{gIoU}_{3D}(b_j, g_i)}{2} \right) \). gIoU \( D \) is known to provide signal even for non-intersecting boxes \([71]\), where the usual IOU \( D \) is always zero. Therefore, we use gIoU \( D \) instead of regular IOU \( D \) for figuring out the best box in 3D as many 3D boxes have a zero IOU \( D \) overlap with the ground truth. For calculating gIoU \( D \), we first calculate the volume \( V \) and hull volume \( V_{hull} \) of the 3D boxes. \( V_{hull} \) is the product of gIoU \( D \) in Birds Eye View (BEV), removing the rotations and hull of the \( Y \) dimension.

\[
\text{gIoU}_{3D}(b_i, b_j) = \frac{V(b_i \cap b_j)}{V(b_i \cup b_j)} + \frac{V(b_i \cup b_j)}{V_{hull}(b_i, b_j)} - 1.
\]

#### Loss Function.

Generally the number of best boxes is less than the number of ground truths in an image, as there could
be some ground truth boxes for which no box is predicted. The tiny number of best boxes introduces a far-heavier skew than the foreground-background classification. Thus, we use the modified AP-Loss [14] as our loss after NMS since AP-Loss does not suffer from class imbalance [14].

Vanilla AP-Loss treats boxes of all images in a mini-batch equally, and the gradients are back-propagated through all the boxes. We remove this condition and rank boxes in an image-wise manner. In other words, if the best boxes are correctly ranked in one image and are not in the second, then the gradients only affect the boxes of the second image. We call this modification of AP-Loss as Image-wise AP-Loss. In other words,

$$\mathcal{L}_{\text{Image-wise}} = \frac{1}{N} \sum_{m=1}^{N} \text{AP}(\mathbf{r}^{(m)}, \text{target}(\mathbf{B}^{(m)})), \quad (11)$$

where $\mathbf{r}^{(m)}$ and $\mathbf{B}^{(m)}$ denote the rescores and the boxes of the $m$th image in a mini-batch respectively. This is different from previous NMS approaches [30–32, 65], which use classification losses. Our ablation studies (Sec. 5.4) show that the Imagewise AP-Loss is better suited to be used after NMS than the classification loss.

Our overall loss function is thus given by $\mathcal{L} = \mathcal{L}_{\text{before}} + \lambda\mathcal{L}_{\text{after}}$ where $\mathcal{L}_{\text{before}}$ denotes the losses before the NMS including classification, 2D and 3D regression as well as confidence losses, and $\mathcal{L}_{\text{after}}$ denotes the loss term after the NMS, which is the Imagewise AP-Loss with $\lambda$ being the weight. See Sec. A2 of the supplementary material for more details of the loss function.

5. Experiments

Our experiments use the most widely used KITTI autonomous driving dataset [25]. We modify the publicly-available PyTorch [59] code of Kinematic-3D [12]. [12] uses DenseNet-121 [34] trained on ImageNet as the backbone and $n_b = 1024$ using 3D-RPN settings of [10]. As [12] is a video-based method while GrooMeD-NMS is an image-based method, we use the best image model of [12] henceforth called Kinematic (Image) as our baseline for a fair comparison. Kinematic (Image) is built on M3D-RPN [10] and uses binning and self-balancing confidence.

Data Splits. There are three commonly used data splits of the KITTI dataset; we evaluate our method on all three.

Test Split: Official KITTI 3D benchmark [1] consists of 7,481 training and 7,518 testing images [25].

Val 1 Split: It partitions the 7,481 training images into 3,712 training and 3,769 validation images [12, 16, 77].

Val 2 Split: It partitions the 7,481 training images into 3,682 training and 3,799 validation images [89].

Training. Training is done in two phases - warm-up and full [12]. We initialize the model with the confidence prediction branch from warmup weights and finetune using the self-balancing loss [12] and Imagewise AP-Loss [14] after our GrooMeD-NMS. See Sec. A3.1 of the supplementary material for more training details. We keep the weight $\lambda$ at 0.05. Unless otherwise stated, we use $p$ as the Linear function (this does not require $\tau$) with $\alpha = 100$. $N_c, \tau$ and $\beta$ are set to 0.4 [10, 12], 0.3 and 0.3 respectively.

Inference. We multiply the class and predicted confidence to get the box’s overall score in inference as in [36, 76, 83]. See Sec. 5.2 for training and inference times.

Evaluation Metrics. KITTI uses AP$_{3D|R_{40}}$ metric to evaluate object detection following [77, 79]. KITTI benchmark evaluates on three object categories: Easy, Moderate and Hard. It assigns each object to a category based on its occlusion, truncation, and height in the image space. The AP$_{3D|R_{40}}$ performance on the Moderate category compares different models in the benchmark [25]. We focus primarily on the Car class following [12].

5.1. KITTI Test 3D Object Detection

Table 2 summarizes the results of 3D object detection and BEV evaluation on KITTI Test Split. The results in Tab. 2 show that GrooMeD-NMS outperforms the baseline M3D-RPN [10] by a significant margin and several other SoTA methods on both the tasks. GrooMeD-NMS also outperforms augmentation based approach MoVi-3D [80] and depth-convolution based D4LCN [22]. Despite being an image-based method, GrooMeD-NMS performs competitively to the video-based method Kinematic (Video) [12], outperforming it on the most-challenging Hard set.

5.2. KITTI Val 1 3D Object Detection

Results. Tab. 3 summarizes the results of 3D object detection and BEV evaluation on KITTI Val 1 Split at two
Table 3: AP_{3D|R_{40}} and AP_{BEV|R_{40}} comparisons on KITTI Val 1 Cars. [Key: Best, Second Best].

<table>
<thead>
<tr>
<th>Method</th>
<th>IoU_{3D} ≥ 0.7</th>
<th>IoU_{3D} ≥ 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP_{3D</td>
<td>R_{40}}(↑)</td>
</tr>
<tr>
<td>MonoDR [5]</td>
<td>12.50 7.34 4.98</td>
<td>- - -</td>
</tr>
<tr>
<td>MonoGRNet [66] in [18]</td>
<td>11.90 7.56 5.76</td>
<td>19.72 12.81 10.15</td>
</tr>
<tr>
<td>MonoDIS [70] in [77]</td>
<td>11.06 7.60 6.37</td>
<td>18.45 12.58 10.66</td>
</tr>
<tr>
<td>M3D-RPN [10] in [28]</td>
<td>14.53 11.07 8.65</td>
<td>20.85 15.62 11.88</td>
</tr>
<tr>
<td>MoVi3D [80]</td>
<td>14.28 11.13 9.68</td>
<td>22.36 17.87 15.73</td>
</tr>
<tr>
<td>MonoPair [18]</td>
<td>16.28 12.30 10.42</td>
<td>24.12 18.17 15.76</td>
</tr>
<tr>
<td>Kinematic (Image) [12]</td>
<td>18.28 13.55 10.13</td>
<td>25.72 18.82 14.48</td>
</tr>
<tr>
<td>Kinematic (Video) [12]</td>
<td>19.76 14.10 10.47</td>
<td>27.83 19.72 15.10</td>
</tr>
<tr>
<td>GrooMeD-NMS (Ours)</td>
<td>19.67 14.32 11.27</td>
<td>27.38 19.75 15.92</td>
</tr>
</tbody>
</table>

Table 4: AP_{3D|R_{40}} and AP_{BEV|R_{40}} comparisons with other NMS on KITTI Val 1 Cars (IoU_{3D} ≥ 0.7). [Key: C= Classical, S= Soft-NMS [8], D= Distance-NMS [76], G= GrooMeD-NMS]

| Method                  | Infer | AP_{3D|R_{40}}(↑) | Easy Mod Hard | AP_{BEV|R_{40}}(↑) | Easy Mod Hard |
|-------------------------|-------|------------------|---------------|------------------|---------------|
| Kinematic (Image)       | C     | 18.28 13.55 10.13 | 25.72 18.82 14.48 | 25.72 18.82 14.48 |
| GrooMeD-NMS             | C     | 19.67 14.31 11.27 | 27.38 19.75 15.93 | 27.38 19.75 15.93 |
| GrooMeD-NMS             | S     | 19.67 14.31 11.27 | 27.38 19.75 15.93 | 27.38 19.75 15.93 |
| GrooMeD-NMS             | D     | 19.67 14.31 11.27 | 27.38 19.75 15.93 | 27.38 19.75 15.93 |
| GrooMeD-NMS             | G     | 19.67 14.31 11.27 | 27.38 19.75 15.93 | 27.38 19.75 15.93 |

Figure 3: Comparison of AP_{3D} at different depths and IoU_{3D} matching thresholds on KITTI Val 1 Split.

IoU_{3D} thresholds of 0.7 and 0.5 [12, 18]. The results in Tab. 3 show that GrooMeD-NMS outperforms the baseline of M3D-RPN [10] and Kinematic (Image) [12] by a significant margin. Interestingly, GrooMeD-NMS (an image-based method) also outperforms the model-based method Kinematic (Video) [12] on most of the metrics. Thus, GrooMeD-NMS performs best on 6 out of the 12 cases (3 categories × 2 tasks × 2 thresholds) while second-best on all other cases. The performance is especially impressive since the biggest improvements are shown on the Moderate and Hard set, where objects are more distant and occluded.

AP_{3D} at different depths and IoU_{3D} thresholds. We next compare the AP_{3D} performance of GrooMeD-NMS and Kinematic (Image) on linear and log scale for objects at different depths of 15, 30, 45, 60 meters and IoU_{3D} matching criteria of 0.3 → 0.7 in Fig. 3 as in [12]. Fig. 3 shows that GrooMeD-NMS outperforms the Kinematic (Image) [12] at all depths and all IoU_{3D} thresholds.

Comparisons with other NMS. We compare with the classical NMS, Soft-NMS [8] and Distance-NMS [76] in Tab. 4. More detailed results are in Tab. 8 of the supplementary material. The results show that NMS inclusion in the training pipeline benefits the performance, unlike [8], which suggests otherwise. Training with GrooMeD-NMS helps because the network gets an additional signal through the GrooMeD-NMS layer whenever the best-localized box corresponding to an object is not selected. Interestingly, Tab. 4 also suggests that replacing GrooMeD-NMS with the classical NMS in inference does not affect the performance.

Score-IoU_{3D} Plot. We further correlate the scores with IoU_{3D} after NMS of our model with two baselines - M3D-RPN [10] and Kinematic (Image) [12] and also the Kinematic (Video) [12] in Fig. 4. We obtain the best correlation of 0.345 exceeding the correlations of M3D-RPN, Kinematic (Image) and, also Kinematic (Video). This proves that including NMS in the training pipeline is beneficial.

Training and Inference Times. We now compare the training and inference times of including GrooMeD-NMS in the pipeline. Warmup training phase takes about 13 hours to train on a single 12 GB GeForce Titan-X GPU. Full training phase of Kinematic (Image) and GrooMeD-NMS takes about 8 and 8.5 hours respectively. The inference time per image using classical and GrooMeD-NMS is 0.12 and 0.15 ms respectively. Tab. 4 suggests that changing the NMS from GrooMeD to classical during inference does not alter the performance. Then, the inference time of our method is the same as 0.12 ms.

5.3. KITTI Val 2 3D Object Detection

Tab. 5 summarizes the results of 3D object detection and BEV evaluation on KITTI Val 2 Split at two IoU_{3D} thresh-
Table 5: AP$_{3D|R_{40}}$ and AP$_{BEV|R_{40}}$ comparisons on KITTI Val 2 Cars. [Key: Best, *= Released, † = Retrained].

<table>
<thead>
<tr>
<th>Method</th>
<th>IoU$_{3D}$ ≥ 0.7</th>
<th>IoU$_{3D}$ ≥ 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP$_{3D</td>
<td>R_{40}}$(†)</td>
</tr>
<tr>
<td></td>
<td>Easy</td>
<td>Mod</td>
</tr>
<tr>
<td>GrooMed-NMS (Ours)</td>
<td>14.72</td>
<td>10.87</td>
</tr>
</tbody>
</table>

Table 6: Ablation studies of our method on KITTI Val 1 Cars.

<table>
<thead>
<tr>
<th>Change from GrooMed-NMS model:</th>
<th>IoU$_{3D}$ ≥ 0.7</th>
<th>IoU$_{3D}$ ≥ 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP$_{3D</td>
<td>R_{40}}$(†)</td>
</tr>
<tr>
<td></td>
<td>Easy</td>
<td>Mod</td>
</tr>
<tr>
<td>Training</td>
<td>16.66</td>
<td>12.10</td>
</tr>
<tr>
<td>Conf+NMS → No Conf+N No NMS</td>
<td>19.16</td>
<td>13.89</td>
</tr>
<tr>
<td>Conf+NMS → No Conf+N No NMS</td>
<td>15.02</td>
<td>11.21</td>
</tr>
<tr>
<td>Initialization</td>
<td>10.33</td>
<td>11.68</td>
</tr>
<tr>
<td>Linear → Exponential, $\tau = 1$</td>
<td>18.63</td>
<td>13.85</td>
</tr>
<tr>
<td>Linear → Exponential, $\tau = 0.5$</td>
<td>18.34</td>
<td>13.79</td>
</tr>
<tr>
<td>Linear → Sigmoid, $\tau = 0.1$</td>
<td>17.40</td>
<td>13.21</td>
</tr>
<tr>
<td>Group+Mask</td>
<td>18.43</td>
<td>13.91</td>
</tr>
<tr>
<td>Group+Mask → Group+No Mask</td>
<td>18.23</td>
<td>13.73</td>
</tr>
<tr>
<td>Loss</td>
<td>16.34</td>
<td>12.74</td>
</tr>
<tr>
<td>Imagewise AP → Vanilla AP</td>
<td>18.26</td>
<td>13.36</td>
</tr>
<tr>
<td>Imagewise AP → BCE</td>
<td>17.51</td>
<td>12.84</td>
</tr>
<tr>
<td>Inference</td>
<td>Class Pred → Class</td>
<td>18.26</td>
</tr>
<tr>
<td>NMS Scores</td>
<td>Class Pred → Pred</td>
<td>17.51</td>
</tr>
<tr>
<td></td>
<td>GrooMed-NMS (best model)</td>
<td>19.67</td>
</tr>
</tbody>
</table>

5.4. Ablation Studies

Tab. 6 compares the modifications of our approach on KITTI Val 1 Cars. Unless stated otherwise, we stick with the experimental settings described in Sec. 5. Using a confidence head (Conf+No NMS) proves beneficial compared to the warmup model (No Conf+No NMS), which is consistent with the observations of [12, 76]. Further, GrooMed-NMS on classification scores (denoted by No Conf+ NMS) is detrimental as the classification scores are not suited for localization [12, 35]. Training the warmup model and then finetuning also works better than training without warmup as in [12] since the warmup phase allows GrooMed-NMS to carry meaningful grouping of the boxes.

As described in Sec. 4.1.5, in addition to Linear, we compare two other functions for pruning function $p$: Exponential and Sigmoid. Both of them do not perform as well as the Linear $p$ possibly because they have vanishing gradients close to overlap of zero or one. Grouping and masking both help our model to reach a better minimum. As described in Sec. 4.3, Imagewise AP loss is better than the Vanilla AP loss since it treats boxes of two images differently. Imagewise AP also performs better than the binary cross-entropy (BCE) loss proposed in [30–32, 65]. Using the product of self-balancing confidence and classification scores instead of using them individually as the scores to the NMS in inference is better, consistent with [36, 76, 83]. Class confidence performs worse since it does not have the localization information while the self-balancing confidence (Pred) gives the localization without considering whether the box belongs to foreground or background.

6. Conclusions

In this paper, we present and integrate GrooMed-NMS—a novel Grouped Mathematically Differentiable NMS for monocular 3D object detection, such that the network is trained end-to-end with a loss on the boxes after NMS. We first formulate NMS as a matrix operation and then do unsupervised grouping and masking of the boxes to obtain a simple closed-form expression of the NMS. GrooMed-NMS addresses the mismatch between training and inference pipelines and, therefore, forces the network to select the best 3D box in a differentiable manner. As a result, GrooMed-NMS achieves state-of-the-art monocular 3D object detection results on the KITTI benchmark dataset. Although our implementation demonstrates monocular 3D object detection, GrooMed-NMS is fairly generic for other object detection tasks. Future work includes applying this method to tasks such as LiDAR-based 3D object detection and pedestrian detection.
References

[18] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3D object detection using pairwise spatial relationships. In *CVPR*, 2020. 1, 2, 6, 7, 8
[30] Paul Henderson and Vittorio Ferrari. End-to-end training of object class detectors for mean average precision. In *ACCV*, 2019. 6, 8
[31] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. A convnet for non-maximum suppression. In *GCPR*, 2016. 2, 5, 6, 8
[33] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3D object detection. In *CVPR*, 2020. 2
[35] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. EPNet: Enhancing point features with image semantics for 3D object detection. In *ECCV*, 2020. 1, 2, 8


[58] Frank Moosmann, Oliver Pink, and Christoph Stiller. Segmentation of 3D LiDAR data in non-flat urban environments using a local convexity criterion. In Intelligent Vehicles Symposium, 2009.


[91] Jianming Zhang, Stan Sclaroff, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Unconstrained salient ob-

---

8983