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

Supplementary Material

A1. Detailed Explanation of NMS as a Matrix Operation

The rescoring process of the classical NMS is greedy set-based [65] and calculates the rescore for a box i (Line 10 of Alg. 1) as

$$r_i = s_i \prod_{j \in \mathbf{d}_{<i}} \left(1 - p(o_{ij})\right), \tag{12}$$

where $\mathbf{d}_{<i}$ is defined as the box indices sampled from $\mathbf{d}$ having higher scores than box i. For example, let us consider that $\mathbf{d} = \{1, 5, 7, 9\}$. Then, for $i = 7$, $\mathbf{d}_{<i} = \{1, 5\}$ while for $i = 1, \mathbf{d}_{<i} = \phi$ with ϕ denoting the empty set. This is possible since we had sorted the scores $\mathbf{s}$ and $\mathbf{O}$ in decreasing order (Lines 2-3 of Alg. 2) to remove the non-differentiable hard argmax operation of the classical NMS (Line 6 of Alg. 1).

Classical NMS only takes the overlap with unsuppressed boxes into account. Therefore, we generalize (12) by accounting for the effect of all (suppressed and unsuppressed) boxes as

$$r_i = s_i \prod_{j=1}^{i-1} \left(1 - p(o_{ij}) r_j\right). \tag{13}$$

The presence of r_j on the RHS of (13) prevents suppressed boxes $r_j \approx 0$ from influencing other boxes hugely. Let us say we have a box b_2 with a high overlap with an unsuppressed box b_1. The classical NMS with a threshold pruning function assigns $r_2 = 0$ while (13) assigns r_2 a small non-zero value with a threshold pruning.

Although (13) keeps $r_i \geq 0$, getting a closed-form recursion in $\mathbf{r}$ is not easy because of the product operation. To get a closed-form recursion with addition/subtraction in $\mathbf{r}$, we first carry out the polynomial multiplication and then ignore the higher-order terms as

$$\begin{aligned} r_i &= s_i \left(1 - \sum_{j=1}^{i-1} p(o_{ij}) r_j + \mathcal{O}(n^2)\right) \\ &\approx s_i \left(1 - \sum_{j=1}^{i-1} p(o_{ij}) r_j\right) \\ &\approx s_i - \sum_{j=1}^{i-1} p(o_{ij}) r_j. \end{aligned} \tag{14}$$

Dropping the s_i in the second term of (14) helps us get a cleaner form of (19). Moreover, it does not change the nature of the NMS since the subtraction keeps the relation $r_i \leq s_i$ intact as $p(o_{ij})$ and r_j are both between $[0, 1]$.

We can also reach (14) directly as follows. Classical NMS suppresses a box which has a high IoU_{2D} overlap with *any* of the unsuppressed boxes ($r_j \approx 1$) to zero. We consider *any* as a logical non-differentiable OR operation and use logical OR $\bigvee$ operator's differentiable relaxation as $\sum$ [38, 47]. We next use this relaxation with the other expression $\mathbf{r} \leq \mathbf{s}$.

When a box shows overlap with more than two unsuppressed boxes, the term $\sum_{j=1}^{i-1} p(o_{ij}) r_j > 1$ in (14) or when a box shows high overlap with one unsuppressed box, the term $s_i < p(o_{ij}) r_j$. In both of these cases, $r_i < 0$. So, we lower bound (14) with a $\max$ operation to ensure that $r_i \geq 0$. Thus,

$$r_i \approx \max\left(s_i - \sum_{j=1}^{i-1} p(o_{ij}) r_j,\ 0\right). \tag{15}$$

We write the rescores $\mathbf{r}$ in a matrix formulation as

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_n \end{bmatrix} \approx \max\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right), \tag{16}$$

with

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_n \end{bmatrix} - \begin{bmatrix} 0 & 0 & \dots & 0 \\ p(o_{21}) & 0 & \dots & 0 \\ p(o_{31}) & p(o_{32}) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ p(o_{n1}) & p(o_{n2}) & \dots & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_n \end{bmatrix}. \tag{17}$$

We next write the above two equations compactly as

$$\mathbf{r} \approx \max(\mathbf{s} - \mathbf{P}\mathbf{r}, \mathbf{0}), \tag{18}$$

where $\mathbf{P}$, called the Prune Matrix, is obtained by element-wise operation of the pruning function p on $\mathbf{O}_{\triangle}$. Maximum

Table 7: Results on using Oracle NMS scores on $\text{AP}_{3\text{D}|R_{40}}$,$\text{AP}_{\text{BEV}|R_{40}}$ and$\text{AP}_{2\text{D}|R_{40}}$ of KITTI Val 1 Cars. [Key: **Best**]

NMS Scores	$\text{AP}_{3\text{D}\|R_{40}}(\uparrow)$			$\text{AP}_{\text{BEV}\|R_{40}}(\uparrow)$			$\text{AP}_{2\text{D}\|R_{40}}(\uparrow)$		
	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
Kinematic (Image)	18.29	13.55	10.13	25.72	18.82	14.48	93.69	84.07	67.14
Oracle $\text{IoU}_{2\text{D}}$	9.36	9.93	6.40	12.27	10.43	8.72	**99.18**	**95.66**	**85.77**
Oracle $\text{IoU}_{3\text{D}}$	**87.93**	**73.10**	**60.91**	**93.47**	**83.61**	**71.31**	80.99	78.38	67.66

operation makes (18) non-linear [41] and, thus, difficult to solve.

However, for a differentiable NMS layer, we need to avoid the recursion. Therefore, we first solve (18) assuming the $\max$ operation is not present which gives us the solution $\mathbf{r} \approx (\mathbf{I}+\mathbf{P})^{-1}\mathbf{s}$. In general, this solution is not necessarily bounded between 0 and 1. Hence, we clip it explicitly to obtain the approximation

$$\mathbf{r} \approx \left\lfloor (\mathbf{I}+\mathbf{P})^{-1}\mathbf{s} \right\rceil, \tag{19}$$

which we use as the solution to (18).

A2. Loss Functions

We now detail out the loss functions used for training. The losses on the boxes before NMS, $\mathcal{L}_{before}$, is given by [12]

$$\begin{aligned} \mathcal{L}_{before} &= \mathcal{L}_{class} + \mathcal{L}_{2\text{D}} + b_{conf}\,\mathcal{L}_{3\text{D}} \\ &\quad + \lambda_{conf}(1-b_{conf}), \end{aligned} \tag{20}$$

where

$$\mathcal{L}_{class} = \text{CE}(b_{class}, g_{class}), \tag{21}$$

$$\mathcal{L}_{2\text{D}} = -\log(\text{IoU}(b_{2\text{D}}, g_{2\text{D}})), \tag{22}$$

$$\begin{aligned} \mathcal{L}_{3\text{D}} &= \text{Smooth-L1}(b_{3\text{D}}, g_{3\text{D}}) \\ &\quad + \lambda_a \text{CE}([b_{\theta_a}, b_{\theta_h}], [g_{\theta_a}, g_{\theta_h}]). \end{aligned} \tag{23}$$

b_{conf} is the predicted self-balancing confidence of each box b, while b_{θ_a} and b_{θ_h} are its orientation bins [12]. g denotes the ground-truth. λ_{conf} is the rolling mean of most recent $\mathcal{L}_{3\text{D}}$ losses per mini-batch [12], while λ_a denotes the weight of the orientation bins loss. CE and Smoooth-L1 denote the Cross Entropy and Smooth L1 loss respectively. Note that we apply 2D and 3D regression losses as well as the confidence losses only on the foreground boxes.

As explained in Sec. 4.3, the loss on the boxes after NMS, $\mathcal{L}_{after}$, is the Imagewise AP-Loss, which is given by

$$\mathcal{L}_{after} = \mathcal{L}_{Imagewise} = \frac{1}{N}\sum_{m=1}^{N} \text{AP}(\mathbf{r}^{(m)}, \text{target}(\mathcal{B}^{(m)})), \tag{24}$$

Let λ be the weight of the $\mathcal{L}_{after}$ term. Then, our overall loss function is given by

$$\mathcal{L} = \mathcal{L}_{before} + \lambda\mathcal{L}_{after} \tag{25}$$

$$\begin{aligned} &= \mathcal{L}_{class} + \mathcal{L}_{2\text{D}} + b_{conf}\,\mathcal{L}_{3\text{D}} + \lambda_{conf}(1-b_{conf}) \\ &\quad + \lambda\mathcal{L}_{Imagewise} \end{aligned} \tag{26}$$

$$\begin{aligned} &= \text{CE}(b_{class}, g_{class}) - \log(\text{IoU}(b_{2\text{D}}, g_{2\text{D}})) \\ &\quad + b_{conf}\ \text{Smooth-L1}(b_{3\text{D}}, g_{3\text{D}}) \\ &\quad + \lambda_a\ b_{conf}\ \text{CE}([b_{\theta_a}, b_{\theta_h}], [g_{\theta_a}, g_{\theta_h}]) \\ &\quad + \lambda_{conf}(1-b_{conf}) + \lambda\mathcal{L}_{Imagewise}. \end{aligned} \tag{27}$$

We keep $\lambda_a = 0.35$ following [12] and $\lambda = 0.05$. Clearly, all our losses and their weights are identical to [12] except $\mathcal{L}_{Imagewise}$.

A3. Additional Experiments and Results

We now provide additional details and results evaluating our system's performance.

A3.1. Training

Training images are augmented using random flipping with probability 0.5 [12]. Adam optimizer [37] is used with batch size 2, weight-decay 5×10^{-4} and gradient clipping of 1 [10, 12]. Warmup starts with a learning rate 4×10^{-3} following a poly learning policy with power 0.9 [12]. Warmup and full training phases take $80k$ and $50k$ mini-batches respectively for Val 1 and Val 2 Splits [12] while take $160k$ and $100k$ mini-batches for Test Split.

A3.2. KITTI Val 1 Oracle NMS Experiments

As discussed in Sec. 1, to understand the effects of an inference-only NMS on 2D and 3D object detection, we conduct a series of oracle experiments. We create an oracle NMS by taking the Val Car boxes of KITTI Val 1 Split from the baseline Kinematic (Image) model *before* NMS and replace their scores with their true $\text{IoU}_{2\text{D}}$ or $\text{IoU}_{3\text{D}}$ with the ground-truth, respectively. Note that this corresponds to the oracle because we do not know the ground-truth boxes during inference. We then pass the boxes with the oracle scores through the classical NMS and report the results in Tab. 7.

The results show that the $\text{AP}_{3\text{D}}$ increases by a staggering $>$ 60 AP on Mod cars when we use oracle $\text{IoU}_{3\text{D}}$ as the NMS score. On the other hand, we only see an increase in $\text{AP}_{2\text{D}}$ by $\approx$ 11 AP on Mod cars when we use oracle $\text{IoU}_{2\text{D}}$ as the NMS score. Thus, the relative effect of using oracle $\text{IoU}_{3\text{D}}$ NMS scores on 3D detection is more significant than using oracle $\text{IoU}_{2\text{D}}$ NMS scores on 2D detection.

Table 8: $\text{AP}_{3\text{D}|R_{40}}$ and $\text{AP}_{\text{BEV}|R_{40}}$ comparisons with other NMS during inference on KITTI Val 1 Cars.

	Inference NMS	$\text{IoU}_{3\text{D}} \geq 0.7$						$\text{IoU}_{3\text{D}} \geq 0.5$					
		$\text{AP}_{3\text{D}\|R_{40}}(\uparrow)$			$\text{AP}_{\text{BEV}\|R_{40}}(\uparrow)$			$\text{AP}_{3\text{D}\|R_{40}}(\uparrow)$			$\text{AP}_{\text{BEV}\|R_{40}}(\uparrow)$		
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
Kinematic (Image) [12]	Classical	18.28	13.55	10.13	25.72	18.82	14.48	54.70	39.33	31.25	60.87	44.36	34.48
Kinematic (Image) [12]	Soft [8]	18.29	13.55	10.13	25.71	18.81	14.48	54.70	39.33	31.26	60.87	44.36	34.48
Kinematic (Image) [12]	Distance [76]	18.25	13.53	10.11	25.71	18.82	14.48	54.70	39.33	31.26	60.87	44.36	34.48
Kinematic (Image) [12]	GrooMeD	18.26	13.51	10.10	25.67	18.77	14.44	54.59	39.25	31.18	60.78	44.28	34.40
GrooMeD-NMS	Classical	19.67	14.31	11.27	27.38	19.75	15.93	55.64	41.08	32.91	61.85	44.98	36.31
GrooMeD-NMS	Soft [8]	19.67	14.31	11.27	27.38	19.75	15.93	55.64	41.08	32.91	61.85	44.98	36.31
GrooMeD-NMS	Distance [76]	19.67	14.31	11.27	27.38	19.75	15.93	55.64	41.08	32.91	61.85	44.98	36.31
GrooMeD-NMS	GrooMeD	19.67	14.32	11.27	27.38	19.75	15.92	55.62	41.07	32.89	61.83	44.98	36.29

In other words, the mismatch is greater between classification and 3D localization compared to the mismatch between classification and 2D localization.

A3.3. KITTI Val 1 3D Object Detection

Comparisons with other NMS. We compare our method with the other NMS—classical, Soft [8] and Distance-NMS [76] and report the detailed results in Tab. 8. We use the publicly released Soft-NMS code and Distance-NMS code from the respective authors. The Distance-NMS model uses the class confidence scores divided by the uncertainty in z (the most erroneous dimension in 3D localization [78]) of a box as the Distance-NMS [76] input. Our model does not predict the uncertainty in z of a box but predicts its self-balancing confidence (the 3D localization score). Therefore, we use the class confidence scores multiplied by the self-balancing confidence as the Distance-NMS input.

The results in Tab. 8 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. Moreover, Tab. 8 suggests that we can replace GrooMeD-NMS with the classical NMS in inference as the performance is almost the same even at $\text{IoU}_{3\text{D}} = 0.5$.

How good is the classical NMS approximation? GrooMeD-NMS uses several approximations to arrive at the matrix solution (19). We now compare how good these approximations are with the classical NMS. Interestingly, Tab. 8 shows that GrooMeD-NMS is an excellent approximation to the classical NMS as the performance does not degrade after changing the NMS in inference.

A3.4. KITTI Val 1 Sensitivity Analysis

There are a few adjustable parameters for the GrooMeD-NMS, such as the NMS threshold N_t, valid box threshold v, the maximum group size α, the weight λ for the $\mathcal{L}_{after}$, and β. We carry out a sensitivity analysis to understand how these parameters affect performance and speed, and how sensitive the algorithm is to these parameters.

Table 9: $\text{AP}_{3\text{D}|R_{40}}$ and $\text{AP}_{\text{BEV}|R_{40}}$ variation with N_t on KITTI Val 1 Cars. [Key: **Best**]

	$\text{AP}_{3\text{D}\|R_{40}}(\uparrow)$			$\text{AP}_{\text{BEV}\|R_{40}}(\uparrow)$		
	Easy	Mod	Hard	Easy	Mod	Hard
$N_t = 0.3$	17.49	13.32	10.54	26.07	18.94	14.61
$N_t = \mathbf{0.4}$	**19.67**	**14.32**	**11.27**	**27.38**	**19.75**	**15.92**
$N_t = 0.5$	19.65	13.93	11.09	26.15	19.15	14.71

Table 10: $\text{AP}_{3\text{D}|R_{40}}$ and $\text{AP}_{\text{BEV}|R_{40}}$ variation with v on KITTI Val 1 Cars. [Key: **Best**]

	$\text{AP}_{3\text{D}\|R_{40}}(\uparrow)$			$\text{AP}_{\text{BEV}\|R_{40}}(\uparrow)$		
	Easy	Mod	Hard	Easy	Mod	Hard
$v = 0.01$	13.71	9.65	7.24	17.73	12.47	9.36
$v = 0.1$	19.37	13.99	10.92	26.95	19.84	15.40
$v = 0.2$	19.65	14.31	11.24	27.35	19.73	15.89
$v = 0.3$	19.67	14.32	11.27	27.38	19.75	15.92
$v = 0.4$	19.67	14.33	11.28	27.38	19.76	15.93
$v = 0.5$	19.67	14.33	11.28	27.38	19.76	15.93
$v = \mathbf{0.6}$	**19.67**	**14.33**	**11.29**	**27.39**	**19.77**	**15.95**

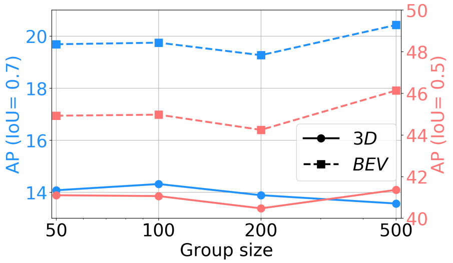

Figure 5: $\text{AP}_{3\text{D}|R_{40}}$ and $\text{AP}_{\text{BEV}|R_{40}}$ Variation with α on Moderate Cars of KITTI Val 1 Split.

Sensitivity to NMS Threshold. We show the sensitivity to NMS threshold N_t in Tab. 9. The results in Tab. 9 show that the optimal $N_t = 0.4$. This is also the N_t in [10, 12].

Sensitivity to Valid Box Threshold. We next show the sensitivity to valid box threshold v in Tab. 10. Our choice of $v = 0.3$ performs close to the optimal choice.

Sensitivity to Maximum Group Size. Grouping has a parameter group size (α). We vary this parameter and report $\text{AP}_{3\text{D}|R_{40}}$ and $\text{AP}_{\text{BEV}|R_{40}}$ at two different $\text{IoU}_{3\text{D}}$ thresholds on Moderate Cars of KITTI Val 1 Split in Fig. 5. We note that the best $\text{AP}_{3\text{D}|R_{40}}$ performance is obtained at $\alpha = 100$ and we, therefore, set $\alpha = 100$ in our experiments.

Sensitivity to Loss Weight. We now show the sensitivity to loss weight λ in Tab. 11. Our choice of $\lambda = 0.05$ is

Table 11: $\text{AP}_{3\text{D}|R_{40}}$ and $\text{AP}_{\text{BEV}|R_{40}}$ variation with λ on KITTI Val 1 Cars. [Key: **Best**]

	$\text{AP}_{3\text{D}|R_{40}}$ (↑)			$\text{AP}_{\text{BEV}|R_{40}}$ (↑)		
	Easy	Mod	Hard	Easy	Mod	Hard
$\lambda = 0$	19.16	13.89	10.96	27.01	19.33	14.84
$\lambda = \mathbf{0.05}$	**19.67**	**14.32**	**11.27**	**27.38**	**19.75**	**15.92**
$\lambda = 0.1$	17.74	13.61	10.81	25.86	19.18	15.57
$\lambda = 1$	10.08	7.26	6.00	14.44	10.55	8.41

Table 12: $\text{AP}_{3\text{D}|R_{40}}$ and $\text{AP}_{\text{BEV}|R_{40}}$ variation with β on KITTI Val 1 Cars. [Key: **Best**]

	$\text{AP}_{3\text{D}|R_{40}}$ (↑)			$\text{AP}_{\text{BEV}|R_{40}}$ (↑)		
	Easy	Mod	Hard	Easy	Mod	Hard
$\beta = 0.1$	18.09	13.64	10.21	26.52	19.50	15.74
$\beta = \mathbf{0.3}$	**19.67**	**14.32**	**11.27**	**27.38**	**19.75**	**15.92**
$\beta = 0.4$	18.91	14.02	11.15	27.11	19.64	15.90
$\beta = 0.5$	18.49	13.66	10.96	27.01	19.47	15.79

the optimal value.

Sensitivity to Best Box Threshold. We now show the sensitivity to the best box threshold β in Tab. 12. Our choice of $\beta = 0.3$ is the optimal value.

Conclusion. Our method has minor sensitivity to N_t, α, λ and β, which is common in object detection. Our method is not as sensitive to v since it only decides a box's validity. Our parameter choice is either at or close to the optimal. The inference speed is only affected by α. Other parameters are used in training or do not affect inference speed.

A3.5. Qualitative Results

We next show some qualitative results of models trained on KITTI Val 1 Split in Fig. 6. We depict the predictions of GrooMeD-NMS in image view on the left and the predictions of GrooMeD-NMS, Kinematic (Image) [12], and ground truth in BEV on the right. In general, GrooMeD-NMS predictions are more closer to the ground truth than Kinematic (Image) [12].

A3.6. Demo Video of GrooMeD-NMS

We next include a short demo video of our GrooMeD-NMS model trained on KITTI Val 1 Split. We run our trained model independently on each frame of the three KITTI raw [24] sequences - 2011_10_03_DRIVE_0047, 2011_09_29_DRIVE_0026 and 2011_09_26_DRIVE_0009. None of the frames from these three raw sequences appear in the training set of KITTI Val 1 Split. We use the camera matrices available with the raw sequences but do not use any temporal information. Overlaid on each frame of the raw input videos, we plot the projected 3D boxes of the predictions and also plot these 3D boxes in the BEV. We set the frame rate of this demo at 10 fps. The demo is also available in HD at `https://www.youtube.com/watch?v=PWctKkyWrno`. In the demo video, notice that the orientation of the boxes are stable despite not using any temporal information.

Acknowledgements

This research was partially sponsored by Ford Motor Company and the Army Research Office under Grant Number W911NF-18-1-0330. This document's views and conclusions are those of the authors and do not represent the official policies, either expressed or implied, of the Army Research Office or the U.S. Government.

We thank Mathieu Blondel and Quentin Berthet from Google Brain, Paris, for several useful discussions on differentiable ranking and sorting. We also discussed the logical operators' relaxation with Ashim Gupta from the University of Utah. Armin Parchami from Ford Motor Company suggested the learnable NMS paper [32]. Enrique Corona and Marcos Paul Gerardo Castro from Ford Motor Company provided feedback during the development of this work. Shengjie Zhu from the Computer Vision Lab at Michigan State University proof-read our manuscript and suggested several changes. We also thank Xuepeng Shi from University College London for sharing the Distance-NMS [76] code for bench-marking. We finally acknowledge anonymous reviewers for their feedback that helped in shaping the final manuscript.

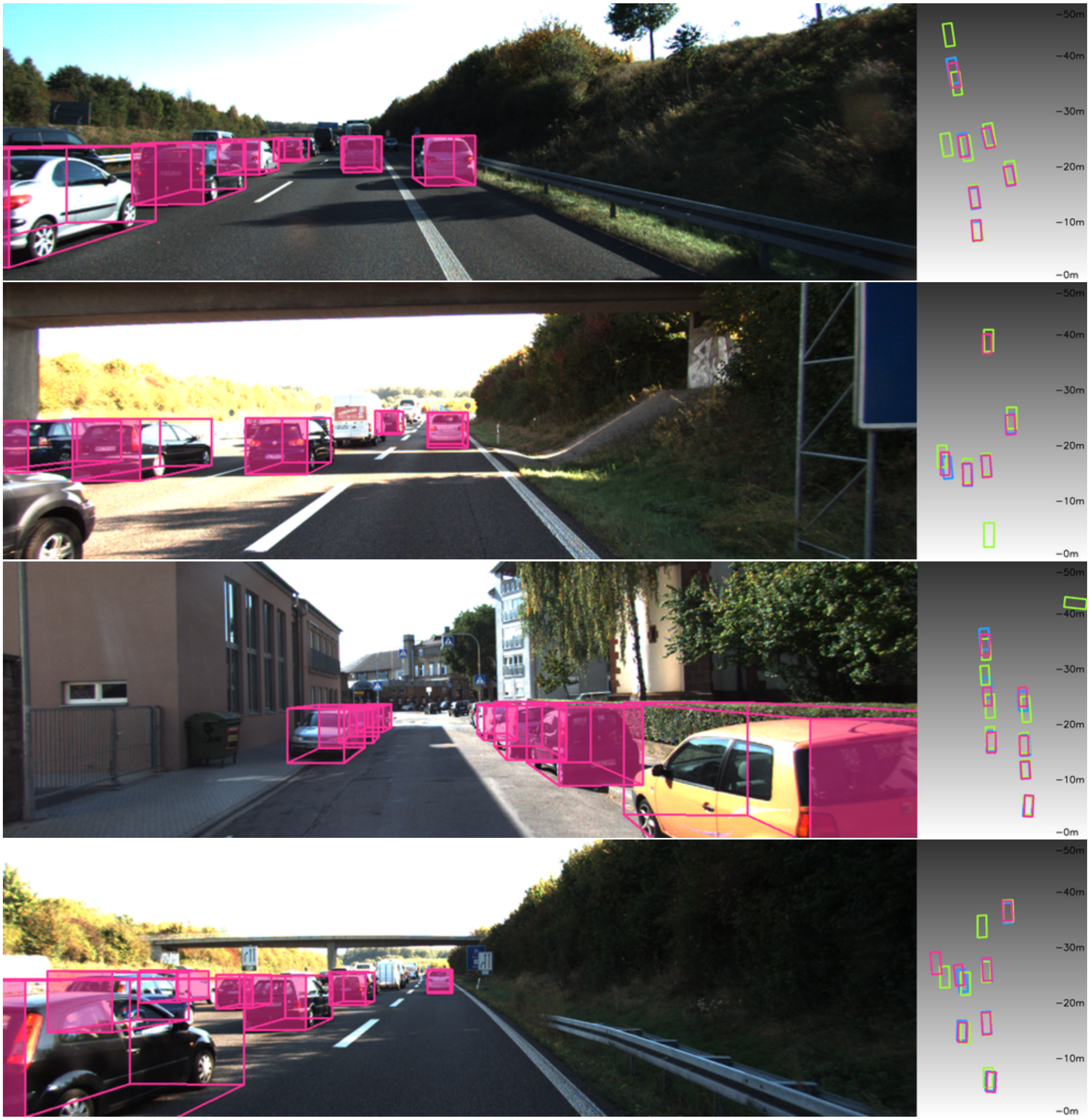

Figure 6: **Qualitative Results** (Best viewed in color). We depict the predictions of GrooMeD-NMS in image view on the left and the predictions of GrooMeD-NMS, Kinematic (Image) [12], and Ground Truth in BEV on the right. In general, GrooMeD-NMS predictions are more closer to the ground truth than Kinematic (Image) [12].