Enhancing Efficiency in 3D Object Detection with Knowledge Distillation

João Correia
Institute for Systems and Robotics
Lisbon, Portugal
joao.diogo.correia@tecnico.ulisboa.pt

Plinio Moreno
Institute for Systems and Robotics
Lisbon, Portugal
plinio@isr.tecnico.ulisboa.pt

***Abstract*—3D object detection is a critical task in various applications today, such as autonomous driving and robotics. Recently, camera-based approaches have gained popularity over LiDAR-based ones due to their cost-effectiveness and more flexible capabilities. Although generally lighter than their LiDAR-based counterparts, almost all state of the art works heavily depend on computationally intensive deep learning methods, which are still quite demanding. We introduce a method for knowledge distillation tailored to the regression problem, aiming to train a significantly lighter student network by considering multiple loss functions for this purpose. Regarding the teacher model, we utilize a bird's-eye-view (BEV) based architecture and propose suggestions for its improvement. We evaluate the proposed methods on the large-scale dataset nuScenes and manage to outperform state of the art works in terms of the trade off between accuracy and computing time.**

***Index Terms*—Knowledge Distillation, 3D Object Detection, BEV models, Autonomous Driving, Teacher-student networks, Deep learning.**

I. Introduction

Understanding objects in three-dimensional space is crucial for a multitude of applications, including autonomous driving and robotics. While LiDAR-based techniques have made significant advancements [1, 2, 3, 4, 5], camera-based methods [6, 7, 8] have gained considerable traction in recent years. In addition to being cost-effective to implement, cameras offer distinct advantages such as the ability to detect distant objects and recognize visual elements on roads such as traffic lights.

Camera-based approaches only have access to the 2D images captured by the cameras. Monocular approaches [9, 10, 11, 12, 8] represent a sizable subset of camera-based approaches. However, the drawback of these approaches lies in their separate processing of individual views, failing to harness information across cameras, thereby resulting in diminished performance.

Instead of relying solely on monocular frameworks, an alternative approach involves adopting a more integrated framework that extracts comprehensive representations from multi-camera images. Models based on BEV [13, 14, 15, 6, 16, 17, 18, 19, 20] have garnered attention within the field of autonomous driving due to their ability to seamlessly merge fragmented raw data from various sensors into a cohesive three-dimensional output space. Typically, a BEV model is constructed with an image backbone, succeeded by a module for view transformation, which elevates perspective image attributes into BEV features. These features are then refined by a BEV feature encoder and specialized heads tailored to specific tasks.

Despite the fact that camera-based approaches are lighter than LiDAR-based systems, deep learning is the conventional tool to detect the objects. Although deep learning-based approaches have achieved notable success, they often require models with a very high number of parameters, sometimes in the order of billions. This increase in needs for computational resources creates an issue for resource-constrained environments, especially when real-time inference is required.

To deal with this problem, one option is to use methods that offload heavy computation to remote systems, using a proper selection method that makes the correct prioritization [21, 22]. Another approach known as Knowledge Distillation (KD) [23] presents a method to transfer knowledge from a large teacher model to a smaller student model. Some works indicate that a compact network trained through KD could achieve comparable accuracy to that of a larger network when subjected to a good optimization process [24]. The main advantage of this would be a significant reduction in the computation resources that are needed, without sacrificing the accuracy of the model.

Most KD works [23, 24, 25] primarily focus on classification. In this use case, KD is effective because of the teacher model's softened logits output. This rich output provides deeper insights than just one-hot encoded class labels. It contains hidden knowledge about the relationships between labels, enhancing the model's ability to generalize. In regression, this does not happen, since the teacher model merely has the same characteristics of the ground truth, with the addition of an unknown error distribution, making KD poorly suited for regression.

To effectively implement KD for regression, a more sophisticated strategy is required. One approach we utilize limits the extent to which the student network can rely on the teacher network. Meanwhile, a more advanced strategy employs the teacher's loss as a confidence score, guiding the student

This work was supported by LARSyS FCT funding (DOI: 10.54499/LA/P/0083/2020, 10.54499/UIDP/50009/2020, and 10.54499/UIDB/50009/2020), by the Lisbon Ellis Unit (LUMLIS) and by the Portuguese Recovery and Resilience Plan (RRP), project number 62, Center for Responsible AI.

network's learning by indicating the reliability of the teacher's predictions.

In this work, we will focus on two main topics. First, we will review some of the main BEV works, and suggest enhancements with the aim of improving performance. We also aim to use knowledge distillation methods in order to create a much more efficient model. Since 3D object detection includes the prediction of bounding boxes, a regression task, and classic knowledge distillation is poorly suited for regression, we will define improved methods for knowledge distillation.

Our contributions include

1) A distilled BEV model for 3D object detection, considerably faster and much more efficient than the original. The knowledge distillation method we present is independent of the use of the BEV model, and can have many more uses beyond 3D object detection;
2) An improvement of the base BEV model architecture, that leads to increased performance in the distilled model.

II. Related Work and Background

A. BEV models

Early research in the field focused on converting camera images to a BEV representation, which is more useful for tasks like object detection. These models transform data from various sensors into a unified 3D representation from above. Orthographic Feature Transform (OFT) [17] was one of the first to do this for single-camera 3D object detection. Pseudo LiDAR [18] took a similar approach by generating a point cloud from a single camera's depth estimates and then working with it in BEV. View Parsing Network (VPN) [6] differed as it combined images from multiple cameras into a top-down view for tasks like semantic segmentation.

More recent techniques benefit from combining features from different sensor perspectives, made possible by advances in 2D to 3D feature transformation. For example, Lift, Splat, Shoot (LSS) [16] improved upon OFT by using a latent depth distribution and pooling features from six images instead of one. Cross-View Transformers (CVT) [19] and Position Embedding Transformation (PETR) [15] took different approaches to integrating perspective and BEV features, with CVT using camera-aware encoding and PETR avoiding explicit BEV feature construction and instead fusing features with 3D positional information directly.

BEVFormer [13] introduced another innovation by using attention mechanisms to transform views and merge features over time, resulting in more flexible BEV features that can handle various sampling grids. It was followed by BevFormer v2 [14], which was designed to work efficiently with modern image backbones. Unlike previous BEV detectors that required depth pre-training and were often limited to specific backbones, BEVFormer v2 focuses on facilitating the optimization process and integrating perspective space supervision to enhance the detection process. They introduce perspective supervision, which addresses the issues of sparse and indirect supervision by providing direct and dense supervision signals. The perspective loss introduced in BEVFormer v2 complements the BEV loss, aiding the optimization of the backbone.

SRCN3D [20] tries to create a more efficient 3D object detection method. The authors remove the dense BEV query mechanism, as they consider it to be computationally inefficient, opting instead for sparse queries, sparse attention with box-wise sampling, and sparse prediction.

Ego3RT [26] uses a polarized grid of "imaginary eyes" and adaptive attention mechanisms, allowing the model to extract rich 3D information from 2D images, achieving superior performance and computational efficiency in standard BEV visual tasks.

B. Knowledge Distillation

Knowledge distillation methods for classification tasks [23, 27, 24, 28, 29, 30, 31, 32] typically depend on the predictions of a teacher network without taking into account any discrepancies between these predictions and the actual truth.

In a typical classification problem, the output layer of a neural network has a softmax function that converts the logits (i.e., the raw output values from the last neural layer) into probabilities. For a given input x, the softmax function σ is applied to the logits z and is defined as:

$$P(y=i|x) = \sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{1}$$

where $P(y=i|x)$ is the probability that the input x belongs to class i, and the sum in the denominator is taken over all possible classes j.

In KD, the knowledge is often transferred by using the softmax function with a temperature parameter T to soften the probabilities. The softened probabilities from the teacher are used to train the student. The softmax function with temperature T is given by:

$$P_T(y=i|x) = \sigma_T(z_i) = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}} \tag{2}$$

The temperature T controls the softness of the probability distribution; a higher value for T produces a softer probability distribution. The student is trained to minimize a loss function that often combines two terms: one that represents the standard cross-entropy with the true labels (hard targets) and another that represents the cross-entropy with the teacher's softened outputs (soft targets). The combined loss function L can be expressed as:

$$L = \alpha \cdot L_{\text{hard}} + (1-\alpha) \cdot L_{\text{soft}} \tag{3}$$

where α is a hyperparameter that balances the importance of the two terms, L_{hard} is the cross-entropy loss with the ground truth, and L_{soft} is the cross-entropy loss with the teacher's softened outputs:

$$L_{\text{hard}} = -\sum_i y_i \log P(y=i|x) \tag{4}$$

$$L_{\text{soft}} = -\sum_i P_T^{\text{teacher}}(y=i|x) \log P_T^{\text{student}}(y=i|x) \quad (5)$$

where y_i is the ground truth (1 for the correct class, 0 for others), $P_T^{teacher}$ is the probability distribution produced by the teacher with temperature T, and $P_T^{student}$ is the probability distribution produced by the student with the same temperature T.

KD can be more effective than just learning from the ground truth because the soft probabilities from the teacher model can provide additional information about the relationships between different classes. For example, the teacher's probabilities might reveal that a picture of a leopard is not only highly probable to be a leopard but also has a non-negligible probability of being a jaguar, providing insights that hard labels do not. This can help the student model learn a richer representation and generalize better to new data.

However, in regression tasks, where predictions are continuous and can vary widely, following the teacher network's guidance without considering its potential errors could mislead the student network. Prior research has attempted to mitigate this problem by using the teacher's predictive error [33] as a constraint, ensuring that the teacher's influence is moderated by its accuracy in relation to the true values.

In the work in [34], the authors overcome the problem by using the loss from the teacher network as a confidence score to guide the knowledge transfer process. The proposed method introduces two new concepts: Attentive Imitation Loss (AIL) and Attentive Hint Training (AHT). AIL uses the teacher's loss as a confidence measure to determine the importance of the teacher's predictions during training. AHT focuses on learning the intermediate representations of the teacher network. The combination of these approaches enables the student network to learn effectively from the teacher network. The results of the paper show promising outcomes. By applying their method, the authors achieved a significant reduction in the number of parameters and computation time while keeping the student network's predictions close to those of the teacher network. The effectiveness of the method was validated on the KITTI dataset.

III. Methodology

A. BEV Model Improvements

As a starting point, we lean on the BEVFormer v2 architecture [14]. BEVFormer v2 is primarily composed of five key elements: a backbone for image processing, a head for perspective 3D detection, an encoder for spatial information, an encoder for temporal data, and a head dedicated to detection in bird's-eye view. This work builds upon the original BEVFormer [13], with changes being made for nearly all components.

Changes to the backbone and the head for perspective 3D detection will be discussed in the following sections, as they will be paramount to the distillation steps. In this section, we focus on architecture enhancements independent of the distillation process, specifically, in regards to the encoders.

Both the spatial and temporal encoders are dependent on BEV Queries $Q \in \mathbb{R}^{H\times W\times C}$ where H, W are the spatial shape of the BEV plane. These are learnable parameters, found during the model training process. In the original architecture, Q is defined only once, shared by both encoders, and learned jointly.

The change we propose to make is separating the queries into two groups Q_s and Q_t, both of the same dimension, for each of the spatial and temporal encoders. These encoders are inherently different parts of the architecture, dealing with different size inputs that represent different entities. By not being restrained to fit both cases, the model will have a higher degree of flexibility and potentially better performance.

B. Knowledge Distillation for Regression Tasks

As discussed before, using Knowledge Distillation for regression tasks in less common, as this method is inherently more powerful for classification use cases. In this work, we will consider multiple options for the teacher-student loss function in order to successfully use this method for a regression case.

In the general regression case, defining as $\hat{\mathbf{y}}_S$ the predictions of the student network, $\hat{\mathbf{y}}_T$ the predictions of the teacher network, and $\mathbf{y}$ as the ground truth labels, the first option:

$$L_{\text{reg}} = \frac{1}{n}\sum_{i=1}^{n} \alpha \left\|\hat{\mathbf{y}}_S - \mathbf{y}\right\| + (1-\alpha)\left\|\hat{\mathbf{y}}_S - \hat{\mathbf{y}}_T\right\|, \quad (6)$$

is similar to the original formulation (3) for classification problems, where α is a balancing factor between the two components.

In practice however, the teacher network can give very erroneous guidance to the student network, sometimes completely inconsistent to the ground truth. A possible way to minimize the influence of these cases is to consider an upper bound on the teacher loss [33]:

$$L_{\text{reg}} = \frac{1}{n}\sum_{i=1}^{n} \alpha \left\|\hat{\mathbf{y}}_S - \mathbf{y}\right\| + (1-\alpha) L_{\text{limit}} \quad (7)$$

$$L_{\text{limit}} = \begin{cases} \left\|\hat{\mathbf{y}}_S - \hat{\mathbf{y}}_T\right\|, & \text{if } \left\|\hat{\mathbf{y}}_S - \mathbf{y}\right\| > \left\|\hat{\mathbf{y}}_T - \mathbf{y}\right\| \\ 0 & \text{otherwise} \end{cases}. \quad (8)$$

A more advanced loss function is introduced in [34]. The main idea is to dynamically weight down teacher predictions that are not as reliable, using for this end, the empirical error of the teacher network with respect to the ground truth.

The loss is defined by the following equations:

$$L_{\text{reg}} = \frac{1}{n}\sum_{i=1}^{n} \alpha \left\|\hat{\mathbf{y}}_S - \mathbf{y}\right\| + (1-\alpha)\Phi_i \left\|\hat{\mathbf{y}}_S - \hat{\mathbf{y}}_T\right\|_i \quad (9)$$

$$\Phi_i = \left(1 - \frac{\left\|\hat{\mathbf{y}}_T - \mathbf{y}\right\|_i}{\eta}\right) \quad (10)$$

$$\eta = \max(e_T) - \min(e_T) \quad (11)$$

$$e_T = \left\{\left\|\hat{\mathbf{y}}_T - \mathbf{y}\right\|_j : j = 1, \ldots, N\right\}, \quad (12)$$

where Φ_i is the normalized teacher loss, e_T is the set of teacher losses of the training set, and η is a normalization parameter.

C. BEV Model Distillation

For the student network, we will conceptualize a lighter version of the original network as defined in III-A.

The backbone is one of the main factors in both computing time and accuracy. Several backbones are considered in the original work, including ResNet [35], DLA [36] VoVNet [37], and InternImage [38]. Most of these backbones are conceptualized with accuracy in mind, often leading to overly heavy computation loads and lower efficiency.

On the other hand, there has also been an effort to develop efficient backbone architectures. MobileNets [39] are a class of efficient models for mobile and embedded vision applications, while [40] proposes guidelines for designing efficient CNNs that go beyond FLOPs to consider direct metrics like speed, influenced by memory access cost and platform characteristics. Finally, the work in [41] offers a systematic approach to scaling CNNs, introducing a new family of models, EfficientNets, which achieve state of the art accuracy with significantly fewer parameters.

However, these EfficientNets are pre-trained on 2D recognition tasks. The work in [14] studies the implication of using backbones trained in this manner for the 3D object detection task, versus using backbones with large-scale depth pre-training. First, there's a significant difference between images from everyday life and those from driving scenes, making it hard for models trained on regular images to understand three-dimensional driving environments, particularly depth. Second, the design of current BEV detectors is quite complex, with the class labels and bounding boxes being separated by multiple transformer layers, leading to a distortion of the gradient flow.

Based on this, it is expected that using EfficientNets by themselves would lead to inferior results. Possible ways to mitigate this issue are also presented, which we can adapt and improve upon.

We use perspective supervision, an extra supervision signal obtained from perspective-view tasks that are applied to the backbone in parallel to the main branch of the model. This parallel branch uses the image details to directly figure out the 3D boxes and categories of objects. The extra supervision signal helps the backbone in learning 3D knowledge, improves the performance of the full model, and unlocks the possibility of using efficient 2D-trained backbones. The error generated by this branch of the model is called the perspective error, and it will be used alongside the traditional error in the final loss calculation.

Aside from the backbone, other parts of the model can also be easily scaled down, in particular, the encoders and the decoder head. When applicable, we scale down by a factor of two the following hyperparameters: the number of layers, and the number of parallel attention heads.

D. Loss Function

As we have seen in the previous section, perspective supervision is critical to make effective use of EfficientNets and to improve the performance of the model in general. This is accomplished by adding an auxiliary perspective loss L_{pers} to the main L_{bev} loss, with the total loss being calculated as

$$L_{\text{total}} = \lambda_{\text{bev}} * L_{\text{bev}} + \lambda_{\text{pers}} * L_{\text{pers}}. \quad (13)$$

In Fig. 1 we present a simplified high level overview of our methods.

IV. Experiments

A. Testing Environment

For both training and testing, we use the nuScenes dataset [42], which contains 1000 annotated videos. Each sample includes 6 camera images and cover the full field of view. The dataset defines a detection score (NDS), a comprehensive and standard metric which combines the mAP (detection accuracy) and five true-positive metrics, namely ATE (translation error), ASE (scale error), AOE (orientation error), AVE (velocity error), and AAE (attribute error). Training and inference are both done in a single RTX3080ti GPU.

B. Experimental Setting

We conduct experiments so that we can understand the effect of the following choices

1) The teacher-student losses for regression;
2) The change in the BEV model architecture - the separation of the queries.

We compare our results to the original BevFormer v2 [14], as well as other recent state of the art methods, namely FCOS3D [8], DETR3D [7], PETR [15], SRCN3D [20] and Ego3RT [26]. In all cases, we consider the performance on the nuScenes test set. For works where multiple model variations are presented, we consider the one with the highest NDS.

We compare both the NDS and the Frames Per Second (FPS) of the algorithms, with the hardware conditions being the same for all trials. FPS refers to the number of images that the algorithm processes in one second, as a measure of the speed of the algorithm. The presented FPS values of the state of the art methods are obtained using the hardware of the testing environment.

We use EfficientNet-B0 as the backbone for our models. We set the loss weights of the BEV loss and perspective loss as $\lambda_{\text{bev}} = \lambda_{\text{pers}} = 1$. For the teacher-student regression losses, we set $\alpha = 0.5$. We use the AdamW [43] optimizer and set the base learning rate as $4e-4$.

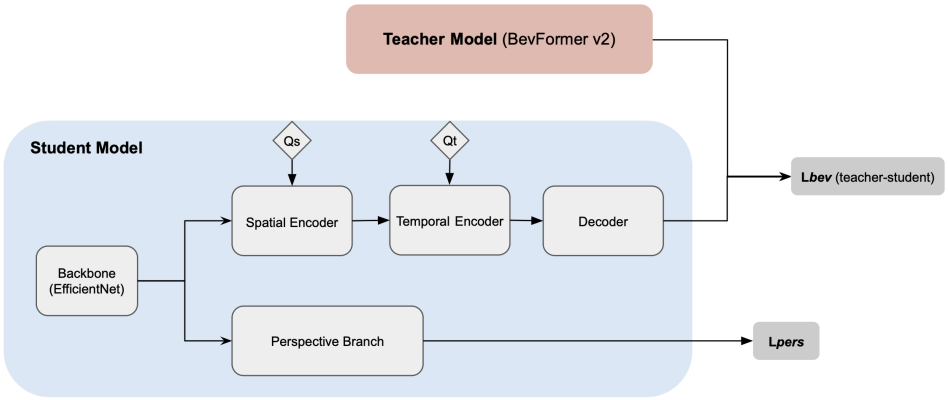

Fig. 1. High level overview of our proposal. On the BEV model architecture itself, we separate the queries for the spatial and temporal encoders, respectively, Q_s and Q_t. The temporal encoder relies on the output from the spatial encoder, both from the current and previous timestamps. We replace the original backbone with a more efficient one - in this case EfficientNet. In order to properly train the backbone, we use perspective supervision, guided by the L_{pers} loss. We use the original BevFormer v2 model as the teacher model, applying knowledge distillation methods guided by the L_{bev} loss.

C. Results

The results are presented in Table I and Fig. 2, comparing the NuScenes Detection Score (NDS) and the Frames Per Second (FPS).

As anticipated, there is a performance decline compared to the original model, but our methods significantly enhance computing efficiency. Regarding the various teacher-student losses for regression, we initially observe that the simpler loss is not very effective for knowledge distillation. Conversely, the regression loss with a limit and with dynamic weighting both perform well, with dynamic weighting achieving superior results.

Concerning the proposed BEV improvement of separating the queries, it indeed results in a modest enhancement in NDS. Providing the model with increased flexibility appears beneficial in this context.

Finally, in comparison with other state of the art methods, our approach demonstrates superior performance in terms of computing time, improving FPS by around 17% compared to the second most efficient method, SRCN3D. In terms of NDS, while there is a performance decline compared to the original model, our best method is quite competitive with others, either on par with or outperforming all of them.

V. Conclusions and Future Work

We introduced modifications to the BEV architecture as well as experiments with various teacher-student loss options for regression. The results demonstrate that more sophisticated regression loss formulations can effectively distill knowledge into a scaled-down version of the original teacher network. This streamlined network significantly improves computing time efficiency, even surpassing recent state of the art methods aimed at the same objective. It also offers greater flexibility in balancing performance versus time, thanks to the availability of numerous backbone options and the ease of scaling the rest of the model. The proposed improvement to the BEV model, namely, the separation of encoder queries, has been shown to enhance performance further.

For future research, several paths could be explored. Firstly, the choice of teacher-student loss is crucial. There is a strong likelihood that we have not yet fully leveraged the information provided by the teacher network. Developing more effective loss functions could yield better results. Regarding the BEV architecture's queries, while separating them has improved performance, these values are still essentially learnable constants. Although this approach is straightforward and has proven effective, it may be more beneficial for these values to be informed or conditioned by relevant input. Lastly, in selecting a scaled-down version of the teacher model, beyond just the backbone, we adopted a relatively simple strategy of reducing some of the most evident hyperparameters across various parts of the architecture. Exploring adjustments to other hyperparameters could be beneficial. More critically, experimenting with more nuanced combinations, employing different scaling factors for different parts of the model, could be advantageous. It is probable that certain components of the model more significantly affect computing time and could represent bottlenecks that might be minimized with minimal impact on performance.

References

[1] Sourabh Vora et al. "Pointpainting: Sequential fusion for 3d object detection". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 4604–4612.

[2] Alex H Lang et al. "Pointpillars: Fast encoders for object detection from point clouds". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 12697–12705.

[3] Yin Zhou and Oncel Tuzel. "Voxelnet: End-to-end learning for point cloud based 3d object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4490–4499.

Method	FPS	NDS (%)
FCOS3D [8]	2.3	42.8
DETR3D [7]	3.0	47.9
PETR [15]	2.8	50.4
SRCN3D [20]	3.6	46.3
Ego3RT [26]	3.5	47.3
Original BevFormer v2 [14]	2.5	**63.4**
Basic regression loss	**4.2**	37.2
Regression loss with limit	**4.2**	46.1
Regression loss with dynamic weight	**4.2**	46.7
Regression loss with dynamic weight + BEV Queries	**4.2**	47.2

TABLE I
RESULTS OF FPS VS NDS WITH VARIOUS LOSSES AND BEV ARCHITECTURES COMPARED TO OTHER STATE OF THE ART WORKS.

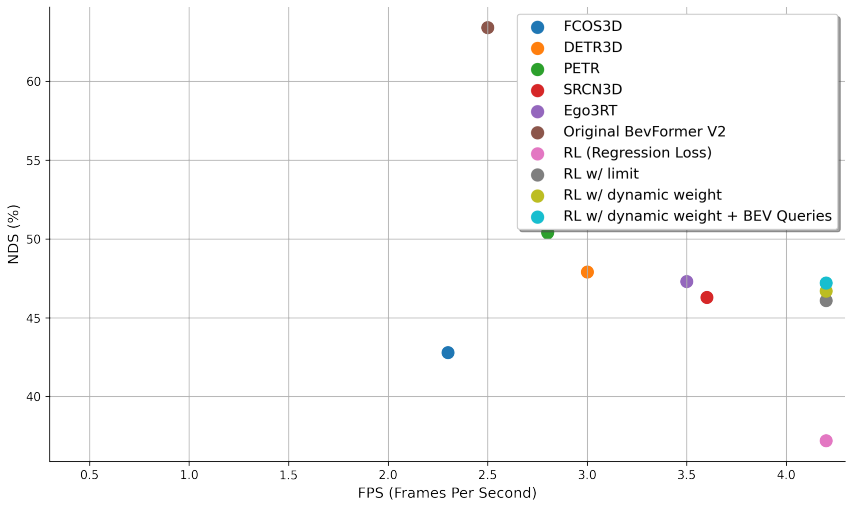

Fig. 2. Scatter plot of the results presented on Table I.

[4] Yan Yan, Yuxing Mao, and Bo Li. "Second: Sparsely embedded convolutional detection". In: *Sensors* 18.10 (2018), p. 3337.

[5] Xiaozhi Chen et al. "Multi-view 3d object detection network for autonomous driving". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 1907–1915.

[6] Bowen Pan et al. "Cross-view semantic segmentation for sensing surroundings". In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 4867–4873.

[7] Yue Wang et al. "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries". In: *Conference on Robot Learning*. PMLR. 2022, pp. 180–191.

[8] Tai Wang et al. "Fcos3d: Fully convolutional one-stage monocular 3d object detection". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 913–922.

[9] Tom Bruls et al. "The right (angled) perspective: Improving the understanding of road scenes using boosted inverse perspective mapping". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 302–309.

[10] Lennart Reiher, Bastian Lampe, and Lutz Eckstein. "A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2020, pp. 1–7.

[11] Dennis Park et al. "Is pseudo-lidar needed for monocular 3d object detection?" In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3142–3152.

[12] Tai Wang et al. "Probabilistic and geometric depth: Detecting objects in perspective". In: *Conference on Robot Learning*. PMLR. 2022, pp. 1475–1485.

[13] Zhiqi Li et al. "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers". In: *European conference on computer vision*. Springer. 2022, pp. 1–18.

[14] Chenyu Yang et al. "BEVFormer v2: Adapting Modern Image Backbones to Bird's-Eye-View Recognition via Perspective Supervision". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 17830–17839.

[15] Yingfei Liu et al. "Petr: Position embedding transformation for multi-view 3d object detection". In: *European Conference on Computer Vision*. Springer. 2022, pp. 531–548.

[16] Jonah Philion and Sanja Fidler. "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer. 2020, pp. 194–210.

[17] Thomas Roddick, Alex Kendall, and Roberto Cipolla. *Orthographic Feature Transform for Monocular 3D Object Detection*. 2018. arXiv: 1811.08188 [cs.CV].

[18] Yan Wang et al. "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8445–8453.

[19] Brady Zhou and Philipp Krähenbühl. "Cross-view transformers for real-time map-view semantic segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 13760–13769.

[20] Yining Shi et al. *SRCN3D: Sparse R-CNN 3D for Compact Convolutional Multi-View 3D Object Detection and Tracking*. 2023. arXiv: 2206.14451 [cs.CV].

[21] João Correia, Alexandre Bernardino, and Ricardo Ribeiro. "Learning Performance Models of Distributed Computer Vision Methods for Decision Making in Detection and Tracking Algorithms in UAVs". In: *IEEE Internet of Things Journal* 10.14 (2023), pp. 12486–12495. DOI: 10.1109/JIOT.2023.3247589.

[22] Bo Yang et al. "Mobile-Edge-Computing-Based Hierarchical Machine Learning Tasks Distribution for IIoT". In: *IEEE Internet of Things Journal* 7.3 (2020), pp. 2169–2180. DOI: 10.1109/JIOT.2019.2959035.

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. arXiv: 1503.02531 [stat.ML].

[24] Adriana Romero et al. "Fitnets: Hints for thin deep nets". In: *arXiv preprint arXiv:1412.6550* (2014).

[25] Tommaso Furlanello et al. "Born again neural networks". In: *International conference on machine learning*. PMLR. 2018, pp. 1607–1616.

[26] Jiachen Lu et al. "Learning ego 3d representation as ray tracing". In: *European Conference on Computer Vision*. Springer. 2022, pp. 129–144.

[27] David Lopez-Paz et al. "Unifying distillation and privileged information". In: *arXiv preprint arXiv:1511.03643* (2015).

[28] Hui Wang et al. "Progressive Blockwise Knowledge Distillation for Neural Network Acceleration." In: *IJCAI*. 2018, pp. 2769–2775.

[29] Junho Yim et al. "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4133–4141.

[30] Vincent Vanhoucke, Andrew Senior, and Mark Z Mao. "Improving the speed of neural networks on CPUs". In: (2011).

[31] Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. "Self-supervised knowledge distillation using singular value decomposition". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 335–350.

[32] Antonio Polino, Razvan Pascanu, and Dan Alistarh. "Model compression via distillation and quantization". In: *arXiv preprint arXiv:1802.05668* (2018).

[33] Guobin Chen et al. "Learning efficient object detection models with knowledge distillation". In: *Advances in neural information processing systems* 30 (2017).

[34] Muhamad Risqi U Saputra et al. "Distilling knowledge from a deep pose regressor network". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 263–272.

[35] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[36] Fisher Yu et al. "Deep layer aggregation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2403–2412.

[37] Youngwan Lee and Jongyoul Park. "Centermask: Real-time anchor-free instance segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 13906–13915.

[38] Wenhai Wang et al. "Internimage: Exploring large-scale vision foundation models with deformable convolutions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 14408–14419.

[39] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[40] Ningning Ma et al. "Shufflenet v2: Practical guidelines for efficient cnn architecture design". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 116–131.

[41] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[42] Holger Caesar et al. "nuscenes: A multimodal dataset for autonomous driving". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11621–11631.

[43] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.