A Novel Look at LIDAR-aided Data-driven mmWave Beam Selection

Matteo Zecchin*, Mahdi Boloursaz Mashhadi*, Mikolaj Jankowski*, Deniz Gündüz, Marios Kountouris, David Gesbert

***Abstract*—Efficient millimeter wave (mmWave) beam selection in vehicle-to-infrastructure (V2I) communication is a crucial yet challenging task due to the narrow mmWave beamwidth and high user mobility. To reduce the search overhead of iterative beam discovery procedures, contextual information from light detection and ranging (LIDAR) sensors mounted on vehicles has been leveraged by data-driven methods to produce useful side information. In this paper, we propose a lightweight neural network (NN) architecture along with the corresponding LIDAR preprocessing, which significantly outperforms previous works. Our solution comprises multiple novelties that improve both the convergence speed and the final accuracy of the model. In particular, we define a novel loss function inspired by the knowledge distillation idea, introduce a curriculum training approach exploiting line-of-sight (LOS)/non-line-of-sight (NLOS) information, and we propose a non-local attention module to improve the performance for the more challenging NLOS cases. Simulation results on benchmark datasets show that, utilizing solely LIDAR data and the receiver position, our NN-based beam selection scheme can achieve 79.9% throughput of an exhaustive beam sweeping approach without any beam search overhead and 95% by searching among as few as 6 beams.**

***Index terms*— mmWave beam selection, LIDAR point cloud, non-local convolutional classifier, curriculum training, knowledge distillation.**

I. Introduction

Millimeter wave (mmWave) communication constitutes a fundamental technology in 5G and future networks, which allows to overcome communication bottlenecks of the over-exploited sub-6GHz bands. To overcome the severe propagation impairments of the above-10GHz spectrum, such as high path attenuation and penetration losses, mmWave communication systems employ massive number of antennas at the base station (BS) to form highly directional beams and attain a large beamforming gain. Because of the narrow mmWave beamwidth, extremely precise alignment and tracking procedures are necessary in order to establish a reliable and high throughput communication link. The optimal communication beam can be easily determined with full channel knowledge; however, in the large antenna regime, obtaining an estimate of the high dimensional channel matrix is costly; and hence, beam selection for efficient communication requires iterative search procedures. In vehicular-to-infrastructure (V2I) communications, for which mmWave communication is envisioned to be a key technology [1], beam selection and tracking are particularly challenging due to the high mobility of the receivers, which leads to reduced beam coherence time [2]. In this scenario, conventional beam selection techniques, such as beam sweeping or multi-level beam selection [3], [4] impose a significant overhead. Therefore, more efficient beam selection techniques that can reduce the cost of iterative search procedure by exploiting contextual information are of great interest.

It has been shown that contextual information from sensors mounted on the vehicles and the infrastructure can be leveraged to reduce the beam selection overhead. For instance, the position information provided by vehicle global positioning system (GPS) can be used to apply an inverse fingerprint approach and query the most prominent mmWave beams [5]. Inertial sensors placed on vehicle's antenna arrays enable efficient antenna element configuration by tracking the orientation of the vehicle [6]. Furthermore, positional and motion information can be jointly processed to further reduce the alignment overhead [7]. From the infrastructure side, a radar located at the BS can help estimate the direction of arrival, which would aid the beam search [8]. Spatial information obtained from out of band measurements was exploited in [9]–[12] where [11], [12] used sub-6GHz channel measurements to train neural network (NN)s for mmWave beamforming. Vision-aided approaches were proposed in [13]–[15]. Base stations equipped with cameras were proposed to employ computer vision and deep learning techniques to predict mmWave blockage and beam strength in [14]. The authors in [15] built a panoramic point cloud from images taken within the cellular coverage area. This point cloud gives a view of the scattering environment, which is then input to a neural network (NN) to predict the optimal beams.

Thanks to recent surge of autonomous driving technologies, high dimensional sensory information is nowadays commonly available also at the vehicle side. For instance, light detection and ranging (LIDAR) is commonly used for autonomous navigation. LIDAR uses a laser to produce a depth map of the environment and surrounding obstacles using delay measurements of the back-scattered signal. Because of the data dimensionality and the lack of analytical models that would relate LIDAR depth map to mmWave beams quality, data-driven methods have been considered to effectively process LIDAR signals as side information for beam search. In [16], [17], a NN architecture was trained over simultaneous LIDAR and ray-tracing channel datasets with a top-k classification metric to identify k beam directions that most probably include the beam resulting in the largest channel gain. In order to reduce the computational cost and NN model size, a simplified classifier architecture that can be trained in a distributed fashion using federated learning was proposed in [18].

This paper builds on the unpublished work of the authors that recently won the "AI/ML in 5G" competition ranking *first* in the "ML for mmWave beam selection" challenge organized by the International Telecommunications Union (ITU) [19], [20]. We propose a convolutional neural network (CNN) architecture along with the corresponding LIDAR preprocessing technique for data-driven mmWave beam selection. The proposed model is trained to exploit LIDAR and positional data in order to identify the best beam directions and reduce the beam search overhead in V2I communication. The specific contributions of this paper in comparison with previous works [16]–[18] can be summarized as follows:

- Inspired by the knowledge distillation (KD) techniques [21], we propose a novel loss function, which not only maximizes the prediction accuracy of the best beam index, but also its correspond-

* Equal contribution.

M. Zecchin, M. Kountouris and D. Gesbert are with the Communication Systems Department, EURECOM, Sophia-Antipolis, France. M. Boloursaz Mashhadi, M. Jankowski and D. Gündüz are with the Dept. of Electrical and Electronic Eng., Imperial College London, UK.

The work of M. Zecchin is founded by the Marie Sklodowska Curie action WINDMILL (grant No. 813999). M. Boloursaz Mashhadi and D. Gunduz received funding from the European Research Council through project BEACON (grant No. 677854). D. Gesbert's and M. Kountouris's contribution are partially supported from a Huawei France-funded Chair towards Future Wireless Networks.

ing power gain. The proposed loss function improves the beam prediction accuracy specifically for smaller k values achieving considerably higher throughput with significantly reduced beam search overhead.

- We utilize a non-local attention scheme, which improves the beam classification accuracy, specifically for the non-of-sight (NLOS) case. Convolutional classifiers used in previous works [16]–[18] learn local features from the LIDAR input and exploit them for beam classification. We observe that the NN utilizing our proposed non-local attention module considerably benefits from a non-local perception of the LIDAR input, specifically in NLOS scenarios where the mmWaves may be reflected from scatterers located far away.
- We propose a curriculum training strategy, which improves both the convergence speed and the final beam prediction accuracy. We observe that for the samples where no dominant LOS component exists, the strongest propagation path becomes significantly less predictable as it depends on the location of scatters and reflectors, which is mainly determined by the traffic conditions. With NLOS samples being more challenging, the proposed curriculum learning starts training with the LOS samples first, and gradually exposes the classifier to more complex NLOS samples. This training strategy achieves faster convergence and improved beam classification accuracy.

Thanks to the above ideas, our NN classifier significantly outperforms previous works [16]–[18]. Utilizing the benchmark Raymobtime dataset [22], [23], our solution achieves top-1, top-5 and top-10 beam selection accuracies of 59.5%, 87.0%, and 92.2%, respectively. In a mmWave communication system with 256 possible beam pairs, our LIDAR-based approach achieves 95% of the available throughput, only by searching among the 6 most probable beams suggested by the NN classifier, greatly reducing the beam search space and the corresponding beam selection overhead. Our classifier harnesses, on average, 79.9% of the available throughput without any beam search at all, just by utilizing the LIDAR and positional side information. Finally, we show that the proposed NN classifier can be further simplified utilizing effective NN pruning techniques without significant loss of its performance while reducing the computational and storage costs for practical deployment.

The content of the paper is organized as follows: Section II introduces the system model. Section III illustrates our proposed NN model for mmWave beam selection utilizing LIDAR data. Simulation results are reported in Section IV. Finally, Section V concludes the paper. The simulation code is publicly available at: https://github.com/MatteoEURECOM/LIDAR-mmWave-Beam-Selection.

II. System Model

We consider a downlink orthogonal frequency-division multiplexing (OFDM) mmWave system using analog beamforming, where the BS located on the street curb serves a vehicle in its coverage area utilizing N_c subcarriers. Both the transmitter and the receiver ends are equipped with antenna arrays with a single radio frequency (RF) chain and fixed complex beam codebooks, which we denote by $\mathcal{C}_t = \{\mathbf{f}_i\}_{i=1}^{C_t}$ and $\mathcal{C}_r = \{\mathbf{w}_j\}_{j=1}^{C_r}$, respectively. The downlink channel matrix from the BS to the vehicle over the n'th subcarrier is denoted by $\mathbf{H}_n$.

For each precoder and combiner vector pair $(i,j) \in \mathcal{C}_t \times \mathcal{C}_r$, the resulting channel gain at subcarrier n is determined by $\mathbf{w}_j^H \mathbf{H}_n \mathbf{f}_i$, where $(\cdot)^H$ denotes the conjugate transpose. We then define the effective power gain matrix $\mathbf{G} \in \mathbb{R}_+^{|\mathcal{C}_t| \times |\mathcal{C}_r|}$, whose (i,j)-th entry contains the aggregate power gain over the N_c subcarries for the transmitter-receiver codebook pair (i,j), as

$$\mathbf{G}_{i,j} = \sum_{n=1}^{N_c} |\mathbf{w}_j^H \mathbf{H}_n \mathbf{f}_i|^2. \tag{1}$$

The optimal pair of precoding and combining vectors is the one that maximizes the channel gain,

$$(i^*, j^*) = \underset{(i,j)}{\operatorname{argmax}}\ \mathbf{G}_{i,j}. \tag{2}$$

Without side information, the transmitter and receiver need to perform an exhaustive search through all $C_t \times C_r$ beam pairs in order to identify (i^*, j^*). Our goal is to infer a small subset of k beam pairs $\mathcal{S}_k \subset \mathcal{C}_t \times \mathcal{C}_r$ exploiting the available position and LIDAR data, such that $(i^*, j^*) \in \mathcal{S}_k$. This results in a reduction of $\frac{k}{C_t \times C_r}$ in the search space of the beam selection procedure. Two metrics to gauge the quality of $\mathcal{S}_k$ as a function of its size k, are the top-k accuracy and top-k throughput ratio. The top-k accuracy is formally defined as

$$A(k) = \mathbb{E}\left[\mathbb{1}\{(i^*, j^*) \in \mathcal{S}_k\}\right], \tag{3}$$

and it measures the fraction of instances for which the best beam index is in the top-k selector output. On the other hand, the top-k throughput ratio is defined as

$$T(k) = \frac{\mathbb{E}\left[\max_{(i,j)\in\mathcal{S}_k} \log_2(1+\mathbf{G}_{i,j})\right]}{\mathbb{E}\left[\log_2(1+\mathbf{G}_{i^*,j^*})\right]} \tag{4}$$

where $\mathbf{G}$ is the effective power matrix defined in (1) and all expectations are with regard to the inherent randomness introduced by vehicles' positions, channel realization, LIDAR and GPS measurements. Note that the top-k throughput ratio is a very informative metric for the problem at hand. In fact, the numerator represents the throughput that can be achieved (at a zero dB transmit SNR) by searching only among the top-k beams suggested by the NN; while the denominator is a normalizing factor representing the maximum throughput achievable by an exhaustive beam sweeping approach.

In the next section, we propose a novel NN architecture that jointly processes positional information along with LIDAR data in order to solve the top-k classification task and therefore, to find the most promising mmWave beams to establish a reliable communication link with reduced beam search overhead.

III. CNN-based Beam Selection Utilizing LIDAR Data

We propose a novel CNN-based beam selection scheme, where connected vehicles utilize measurements from their LIDAR sensor along with their location data to reduce the beam search overhead required to establish a mmWave link with a nearby BS.

A. LIDAR Prepossessing

Raw LIDAR data is in the form of large point cloud measurements $\mathcal{P} = \{(x_i, y_i, z_i)\}_{i=1}^{|\mathcal{P}|}$, where each triplet (x_i, y_i, z_i) represents coordinates of an obstacle point measured by the LIDAR sensor. To avoid excessive computations on large point clouds, we preprocess the raw LIDAR data to get a simplified representation of the coverage area of the BS, which is then input to the classifier CNN. We assume that each vehicle knows the location of the BS and its coverage area, and divides the coverage area into a 2D grid of equal squares. We then produce a top-view representation of the coverage area setting grid entries to 1 whenever at least one point in $\mathcal{P}$ lies within that grid square, and to 0 otherwise. We also embed the location of the vehicle and the BS into this representation by setting the grid value of the square accommodating the BS and the vehicle to -1 and -2, respectively. Fig. 1 shows one such preprocessing step. We note

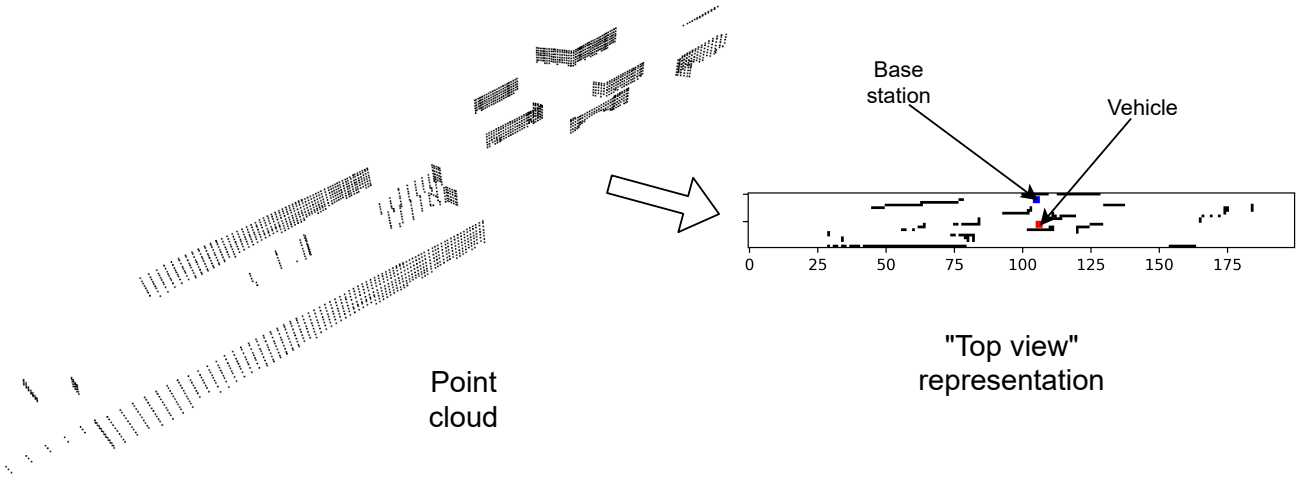


Fig. 1: Preprocessing of the LIDAR point cloud.

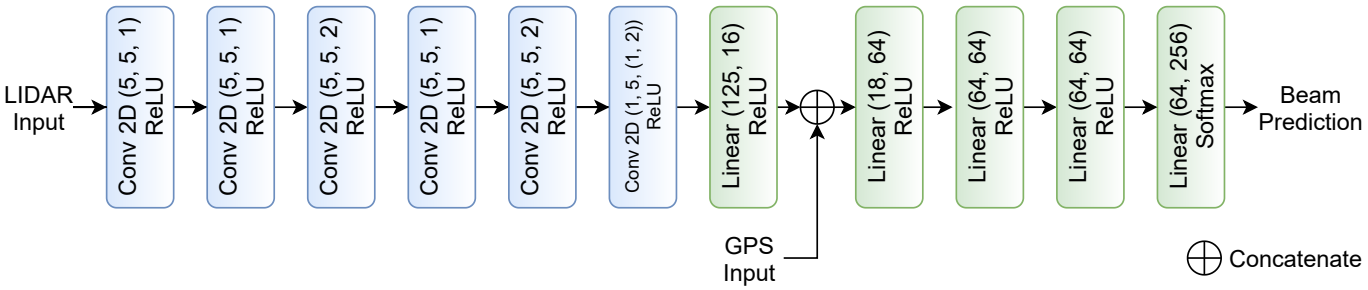


Fig. 2: The proposed CNN model architecture.

that discarding the z-axis causes certain information loss; however, we found this loss not to affect much the accuracy of the beam classification task. On the other hand, it allows us to reduce the complexity of our NN model significantly.

B. NN Architecture

Fig. 2 shows our proposed NN architecture for LIDAR-aided mmWave beam selection, which is composed of 6 convolutional layers and 5 linear layers. Each layer is followed by the rectified linear unit (ReLU) activation, except for the last layer, which is followed by softmax activation to output beam predictions. Our architecture consists of separate branches to process LIDAR and GPS inputs. The LIDAR branch comprises 6 convolutional layers followed by a linear layer to extract features from the preprocessed LIDAR input as in Subsection III-A. The output features from the LIDAR branch are then concatenated with (x, y) location coordinates of the vehicle from the GPS input. We discard the BS coordinates, as these are fixed on the whole dataset and do not need to participate in the training process.

The concatenated feature vector is then input to 4 linear layers. The first three linear layers include 64 neurons and the last one outputs a 256-element vector corresponding to the number of possible beam pairs, i.e., $|C_t| \cdot |C_r| = 256$. Although these linear layers increase the complexity of our NN architecture, in Section III-F we use pruning techniques to effectively reduce the memory and computation requirements of the proposed NN.

C. Loss Function

To define the loss function, we denote by $\mathbf{y} \in \mathbb{R}_+^{(|\mathcal{C}_t| \cdot |\mathcal{C}_r|)}$ the vectorized version of the mmWave power gain matrix $\mathbf{G}$ obtained by the following bijective map

$$\mathbf{y}_{(i-1)\cdot|\mathcal{C}_r|+j} = \mathbf{G}_{i,j}. \tag{5}$$

For each $\mathbf{y}$ vector, we also denote by $\bar{\mathbf{y}}$ the vector $\mathbf{y}$ normalized to unit Euclidean norm and by $\mathbf{y}^*$ the unitary vector that is non-zero on the component corresponding to the largest entry of $\mathbf{y}$ (ties broken arbitrarily). Then, for $\beta \in [0, 1]$, we train our model by minimizing the following loss function

$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = (1 - \beta)\mathcal{H}(\mathbf{y}^*, \hat{\mathbf{y}}) + \beta\mathcal{H}(\bar{\mathbf{y}}, \hat{\mathbf{y}}) \tag{6}$$

where $\hat{\mathbf{y}}$ is the model prediction and $\mathcal{H}(\cdot)$ denotes the empirical cross entropy that, for two non-negative unit norm vectors p and q in $\mathbb{R}^d$ is defined as

$$\mathcal{H}(p, q) = -\sum_{i=1}^{d} p_i \log(q_i). \tag{7}$$

The first term in (6) is a standard multi-class cross entropy loss that enforces the NN to predict the indices of the beam associated to the strongest mmWave beam. The second term, instead, drives the NN to match the output of each neuron in the last layer to the normalized power gain of the corresponding mmWave beam pair. This is achieved by treating $\mathbf{y}$ as a vector of "soft labels" and minimizing the corresponding empirical cross entropy loss. This last term is specifically effective in top-k classification for $k > 1$, where it is advisable not to output exclusively the best beam pair, but also accurately predict k competing candidate beam pairs. Finally, the

value $\beta \in [0, 1]$ provides a tradeoff between the two terms. The idea of combining two training objectives as in Eq. (6) resembles the KD technique [21]. KD is a popular model compression technique that aims at instilling the knowledge of a large classifier network, termed as teacher, into a lightweight student classifier. This is achieved by augmenting the original labels of a dataset by the soft prediction of the teacher model. This additional training objective has been shown to improve the performance of the student model, in some cases even outperforming the teacher model, and to act as a regularization term [21], [24]. In the context of mmWave beam selection, we show experimentally that by exploiting the soft labels, we are able to obtain similar gains and improve the predictive capabilities of the trained model.

D. Curriculum Training with LOS/NLOS Samples

In the absence of LOS, the predictability of the strongest propagation paths greatly decreases as a consequence of the prominent dependency on the relative positions of scatterers and reflectors. However, the presence of neighbouring moving obstacles renders NLOS condition frequent in vehicular type of communication. The difficulty of the prediction task in NLOS scenarios is so high compared to the LOS case that, data-driven methods tend to be biased towards the easier LOS samples to the detriment of the NLOS performance. In order to address this performance imbalance, we propose to adapt the sample distribution during training time so that the challenging instances are less likely during the initial phases of learning. This procedure is motivated by curriculum learning [25], [26], which suggests to expose the training process to easier instances at the initial phases and to gradually increase the difficulty of the tasks. To apply this strategy it is necessary to first define *scoring* and a *pacing* functions: the former assigns a level of difficulty to each sample while the latter determines at which rate the transition should be made from easier samples to harder ones during the learning process.

For the task at hand, a natural measure of difficulty is the absence of LOS and an effective way to modulate the difficulty of the learning task consists in changing the probability of NLOS samples during the training epochs. Hence, denoting by P the feature distribution from which the original training dataset $\mathcal{D}$ is generated, we exploit a biased sampling scheme to generate skewed dataset $\mathcal{D}_\lambda$, whose hardness is proportional to the rejection coefficient $\lambda \in [0, 1]$. In particular, the set of instances $\mathcal{D}_\lambda$ is created from $\mathcal{D}$ by independently removing each NLOS sample with probability $1-\lambda$. As a result, $\mathcal{D}_\lambda$ represents a sample drawn from the following distribution

$$P_\lambda \propto (1-q)P_{LOS} + q\lambda P_{NLOS} \tag{8}$$

where P_{LOS} is the feature distribution conditioned on the presence of LOS, P_{NLOS} the distribution conditioned on its absence, and q is the probability of NLOS condition under the original distribution P. The pacing function is represented by a sequence $\{\lambda_i\}$ that for each epoch i determines the probability that a NLOS sample is accepted for training. As shown below, a properly chosen sequence $\{\lambda_i\}$ can improve both the convergence speed and the accuracy of the final solution compared to the unstructured and randomized sampling of training instances.

E. Non-local Attention for Improved NLOS Performance

The convolutional layers in Fig. 2 learn and exploit local features from the LIDAR input and use them for beam classification. However, the classifier can benefit from non-local perception of the coverage area specifically in the NLOS cases. In order to extract non-local perception of the coverage area from the LIDAR input in an efficient manner, we use a non-local attention module introduced in [27], [28]. We later show through simulations that the non-local attention module further improves the performance. The general input-output relation for non-local attention is given by

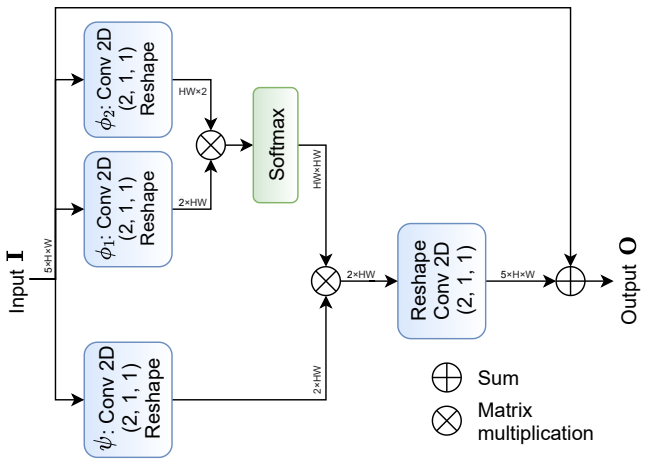


Fig. 3: Block diagram of the non-local attention module.

$$\mathbf{O}_i = \mathbf{I}_i + \frac{1}{\eta(\mathbf{I})} \sum_{\forall j} \phi(\mathbf{I}_i, \mathbf{I}_j)\psi(\mathbf{I}_j), \tag{9}$$

in which i is the space-time index of an output position whose response is to be computed, j is the index that enumerates all possible positions, $\mathbf{I}$ is the input, and $\mathbf{O}$ is the output of the same size as $\mathbf{I}$. A pairwise function ϕ computes a scalar representing the relationship between i and all j. The unary function ψ computes a representation of the input signal at position j. The response is normalized by the factor $\eta(\mathbf{I})$. We tried various popular choices for $\eta(\cdot), \phi(\cdot)$ and $\psi(\cdot)$ functions (refer to [28] for further details), and found that the best performing ones in our case are the embedded Gaussian function for $\phi(\mathbf{I}_i, \mathbf{I}_j)$, given by $\phi(\mathbf{I}_i, \mathbf{I}_j) = \exp\left([\mathbf{W}_{\phi_1}\mathbf{I}_i]^T[\mathbf{W}_{\phi_2}\mathbf{I}_j]\right)$, $\psi(\mathbf{I}_j) = \mathbf{W}_\psi \mathbf{I}_j$, and $\eta(\mathbf{I}) = \sum_{\forall j} \phi(\mathbf{I}_i, \mathbf{I}_j)$, as will be discussed later in Section IV. Here, $\mathbf{W}$s are trainable weight matrices. Fig. 3 provides the block diagram for our non-local attention module, where 1×1 convolutions implement $\mathbf{W}$ weight multiplications, and Softmax activation implements the exponential function. Refer to [28] for more details on attention modules.

F. Network Pruning

To reduce the computational and memory footprint of the proposed model, we introduce an additional network pruning [29], [30] step. Network pruning is a method for reducing the computational complexity and the size of NNs by finding redundant neurons and removing them, based on some saliency measure. In this work we employ a straightforward approach of removing neurons or filters with the lowest L_1-norm of the weights, which already leads to maintaining a satisfactory performance with low network complexity. We leave the exploration of more effective pruning methods for future work. Since the majority of the weights in our network are contained in the later linear layers the convolutional part of the network is already lightly parameterized. In this work we explore two particular strategies for network pruning: *unstructured* and *structured* pruning. Unstructured pruning removes the weights with the smallest magnitude, whereas structured pruning removes entire neurons with the lowest mean magnitude of weights. Unstructured pruning usually leads to better gains, as it is able to prune more parameters, while maintaining satisfactory performance.

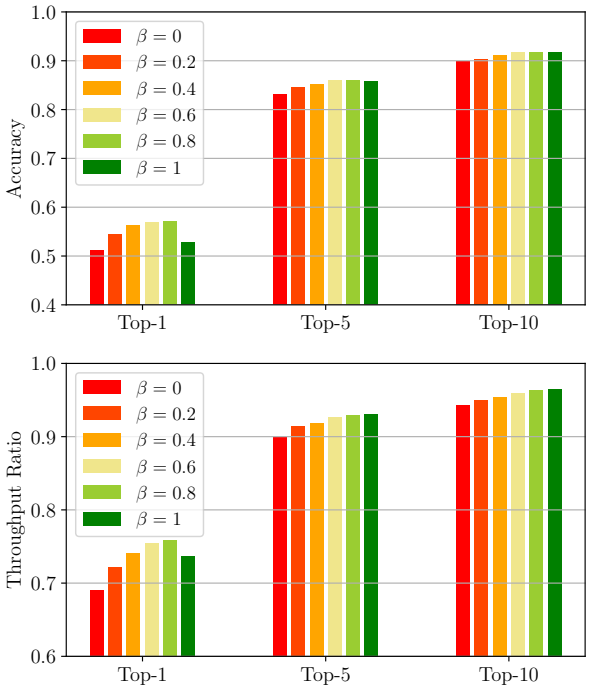


Fig. 4: Top-k accuracy and througput ratio for $k \in \{1, 5, 10\}$ averaged over 10 training runs for different β values.

IV. Numerical Evaluations

In this section, we provide a series of experiments that highlighting the performance improvements and benefits that each of the proposed techniques and architecture designs can attain. Subsequently, we evaluate the proposed solution with all of the above enhancements, compare it against the state of the art and showcase its superiority. All experiments are carried out using the Raymobtime dataset [22], [23], in particular using the s008 portion for training and the s009 portion for testing. To compare the performance of the different solutions, we consider the top-k classification accuracy and the top-k throughput ratio of top-k beam selector $\mathcal{S}_k(\hat{\mathbf{y}})$, which for a network output $\hat{\mathbf{y}}$, returns the set of indices corresponding to the predicted k strongest beams.

A. Choice of β

We optimize the NN parameters by minimizing the proposed loss function (6). Note, however, that the proposed loss function is parameterized by $\beta \in [0, 1]$ weighting the training signal coming from the hard and soft labels, respectively, represented by the one-hot-encoded best beam index $\mathbf{y}^*$ and the normalized channel gain vector $\bar{\mathbf{y}}$. Choosing a proper value for β is necessary in order to strike a good balance between these two pulling forces. For small β values, the objective function resembles the standard multi-class cross-entropy loss and the trained NN tends to act as a myopic classifier that tries to correctly predict the best beam index regardless of the performance of the other beams. On the other extreme, very large values of β drive the output of the model to match the normalized channel gains associated with different beams and to potentially trade the best beam prediction accuracy for this purpose. We evaluate the effect of β by training the model proposed in Sec. III-B for $\beta \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ and we report the final top-k accuracy and throughput ratio averaged over 10 runs for each β value in Fig. 4. We observe that a value of $\beta = 0.8$ yields a trained model that combines the best of the two above mentioned behaviours. In fact, for this choice of β, the predictor attains the highest top-1 accuracy, outperforming even the models trained for smaller β values that prioritize this metric. At the same time, for larger k, its performance is indistinguishable from the network trained with $\beta = 1$. The same conclusion holds for the top-k throughput ratio metric. The model trained with the optimal β value is able to provide 76% top-1 throughput ratio, while the ratios for the two worst performing values are 69% ($\beta = 0$) and 73% ($\beta = 1$).

TABLE I: Performance of curriculum, anti-curriculum and standard training procedures.

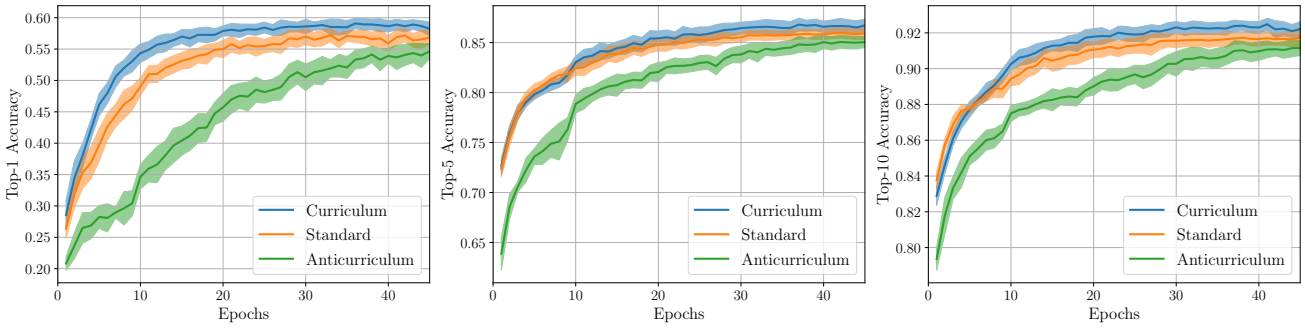
Strategy	$A(1)$	$A(5)$	$T(1)$	$T(5)$
Curr.	**58.1%**	**86.6%**	**77.9%**	**94.1%**
Standard	57.1%	86.0%	75.9%	92.9%
Anti-curr.	53.8%	85.1%	71.9%	91.2%

B. Curriculum Training

While it is usual practice to train NNs using batches of data sampled uniformly at random from the training dataset, in the following we illustrate the benefits of biasing the sampling procedure in order to obtain a sequence of training samples with an increasing level of difficulty. As illustrated in Section III-D, this is achieved by employing a sample rejection strategy based on the presence or absence of LOS components. Specifically, we train the proposed NN architecture for 45 epochs decreasing the NLOS rejection probability $1 - \lambda$ from 1 by steps of 0.2 every 9 epochs until reaches 0. In this manner, the first batches contain only LOS samples, whereas during the last 9 epochs the ratio between LOS and NLOS will be the same as the one in the original unbiased empirical distribution. We also consider the opposite strategy, namely exposing the NN to batches of hard samples first. This is obtained by the same sampling procedure but with the role of NLOS and LOS swapped. We term this alternative anti-curriculum as it starts from the hardest instances. As a natural baseline, we also consider the standard unbiased sampling procedure. For each case, we train the same model architecture proposed in Sec. III-B using the loss function with $\beta = 0.8$. Distinct training dynamics result in different convergence time and final accuracy values. In Fig. 5, we plot the evolution of the accuracy metrics, averaged over 10 repetitions, of the standard, curriculum and anti-curriculum learning procedures. In terms of convergence time the curriculum learning strategy outperforms both the standard and anti-curriculum sampling schemes as it quickly plateaus to higher accuracy levels. The final performance, also averaged over 10 runs, is reported in Table I. The curriculum learning strategy improves by 2% the top-1 throughput ratio and by 1.2% the top-5 throughput ratio compared to standard learning. On the other hand, anti-curriculum learning has a detrimental effect on the performance, resulting in a performance loss of 4% and 1.7% in terms of top-1 and top-5 throughput ratio, respectively.

C. Non-local Attention

In the following we consider augmenting the NN architecture of Sec. III-B by adding one non-local attention module located after the fifth convolutional layer of the LIDAR processing branch. This design choice represents the best trade-off between the performance gain and additional computational burden. In fact, even if multiple non-local blocks generally lead to better performance, our NN is reasonably shallow and it does not display a significant improvement when more



(a) $k = 1$ (b) $k = 5$ (c) $k = 10$

Fig. 5: Evolution of the top-k accuracy over the training epochs for curriculum, standard and anti-curriculum training approaches. The curves are averaged over 10 training runs.

TABLE II: Performance improvement by the proposed non-local attention approach.

$\phi(\cdot)$	$A(1)$	$A(5)$	$T(1)$	$T(5)$
Embedded	**58.3%**	**86.7%**	**78.7%**	**94.3%**
Gaussian	57.9%	86.5%	77.7%	94.3%
Dot	57.7%	86.7%	77.7%	94.1%
Without NLA	57.1%	86.0%	75.9%	92.9%

than one non-local block is instantiated. Additionally, we did not observe any meaningful variation in the final performance by placing the non-local block at different depths in the network. On the other hand, the number of floating point operations (FLOPs) required for a forward pass through the non-local module depends on its position within the NN. In fact, high level representations at deeper stages of the NN have lower dimension resulting in modest computational effort when processed by the non-local block. In particular, the output of the fifth convolutional layer is a tensors of size $5 \times 50 \times 5$, that is further sub-sampled using a max pooling kernel of size $1 \times 2 \times 2$. This operation preserves the non-local behaviour but it translates in computational cost of only 250k floating point operations.

We test three popular choices for the non-local operations denoted by $\phi(\cdot)$: Gaussian, embedded Gaussian and dot product pair-wise functions. For each, we train the model minimizing the proposed loss for $\beta = 0.8$ and the standard sampling procedure. In this way we are able to assess the gain that the non-local attention module brings independently of curriculum learning. We average the final performance over 10 runs and report the average results in Table II. Across the three different choices of $\phi(\cdot)$ we do not observe great variability, with the non-local attention using embedded Gaussian slightly outperforming the others. Nonetheless, compared to the same model without non-local attention we measure an improvement of 2.8% and 1.4% in terms of top-1 and top-5 throughput ratios, respectively.

D. Pruning

We consider network pruning to further reduce the computational and storage cost of deploying the proposed NN solution. Before we prune any of the weights, we first pretrain the network, following the strategy described in Section III-B. After the pretraining is finished, we run multiple iterations of pruning by first discarding a portion of the remaining weights with the lowest L_1-norm and fine-tuning the network by following the same training strategy that we used for pretraining. We evaluate both the unstructured and structured pruning methodologies introduced in Sec. III-F. The results are reported in Fig. 6, where we plot the top-k accuracy and top-k throughput ratio of the pruned model against the pruning ratio. The pruning ratio is defined here as the fraction of the weights removed from the unpruned model.

The unstructured pruning achieves better accuracy and throughput ratio at larger pruning rates, compared to structured pruning. This is due to the fact that unstructured pruning can be more precise in removing weights since it works by removing single weights rather than entire columns or rows of the weight matrix. Nevertheless, in unstructured pruning, weight matrices of the linear layers are only partially sparsified, which is computationally sub-optimal compared to removing entire rows or columns as structured pruning does. In both cases we find that the pruned model maintains excellent predictive performance even for large pruning ratios. In fact, it is possible to obtain a top-10 throughput ratio above 96.5%, despite pruning 60% of the weights in the case of structured pruning, and 95.9% with unstructured pruning. A graceful performance degradation also happens in terms of accuracy. Therefore, we conclude that pruning represents a viable option to further reduce the computational and storage footprint of learned models while maintaining excellent predictive capabilities.

E. Comparison with Previous Works

In the following we evaluate and compare the proposed final solution that includes all the above improvements against two state-of-the-art data-driven approaches. The first baseline model we consider is the 13-layer NN presented in [16], [17] and the second one is the lighter version recently proposed in [18] that comprises 8 layers. In Table III we compare the accuracy and throughput achieved by these models along with their number of trainable parameters and FLOPs. These results are averaged over 10 MonteCarlo rounds of training reported with the corresponding 95% confidence intervals. For a fair comparison, we have reported results for our unpruned architecture. The number of trainable parameters and FLOP count directly relate to the computational and memory footprint of the models, and therefore, are of interest in order to asses their deployment feasibility. Compared to the larger architecture in [16], [17], our model requires only 7.6% of the trainable parameters and 2.5% of the floating point operations to perform the forward pass. At the same time our model is comparable to [18] in terms of computational and storage cost; the difference can be further reduced using the pruning techniques proposed in Sec. III-F. This renders the proposed solution also applicable in distributed training scenarios with memory and computationally constrained devices. Our model consistently outperforms both baselines in term

TABLE III: Performance comparison between the proposed NN model and previous works.

Model	$A(1)$	$T(1)$	$A(5)$	$T(5)$	$A(10)$	$T(10)$	FLOP count	# params.
[16], [17]	$31.5 \pm 2.6\%$	$46.1 \pm 2.6\%$	$71.9 \pm 2.2\%$	$76.1 \pm 1.9\%$	$83.9 \pm 0.9\%$	$86.1 \pm 0.8\%$	179.01×10^6	403677
[18]	$52.3 \pm 1.9\%$	$70.3 \pm 2.6\%$	$85.3 \pm 0.9\%$	$90.8 \pm 1.5\%$	$91.1 \pm 0.3\%$	$94.7 \pm 0.6\%$	1.72×10^6	7462
Proposed	$59.5 \pm 0.5\%$	$79.9 \pm 0.8\%$	$87.0 \pm 0.3\%$	$94.6 \pm 0.8\%$	$92.2 \pm 0.2\%$	$96.9 \pm 0.6\%$	4.55×10^6	30872

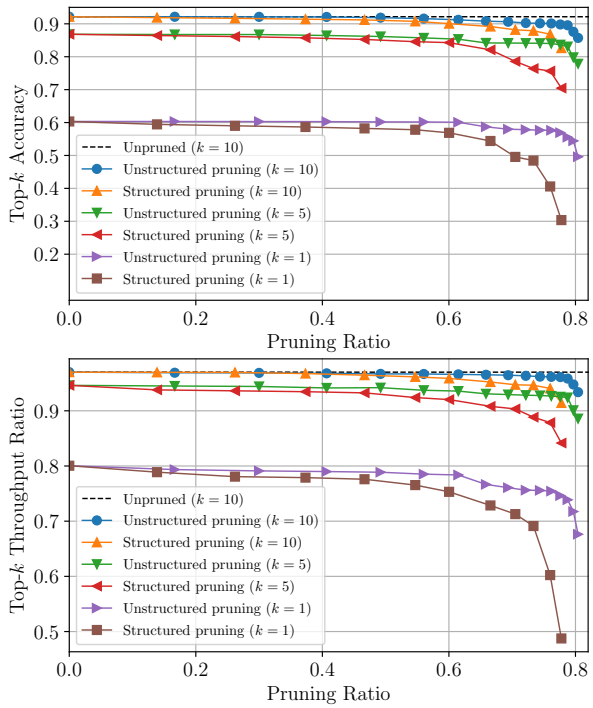


Fig. 6: Accuracy and throughput ratio as a function of pruning ratio.

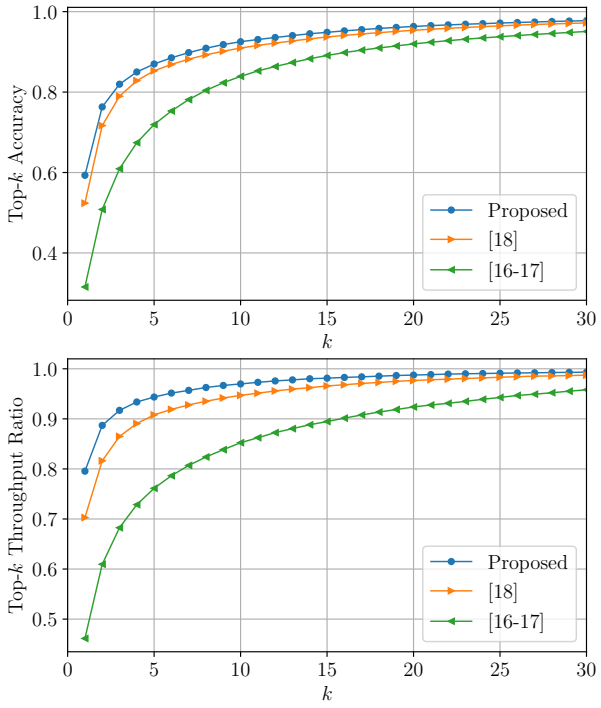


Fig. 7: Accuracy and throughput ratio performance curves comparing the proposed solution and previous work.

of top-k accuracy and throughput ratio metrics, which are reported in Fig. 7 for $k \in [0, 30]$. Our model yields a striking 79.9% top-1 throughput ratio, harnessing a great portion of the available rate without any search procedure. The two competing baselines attain only 46.1% and 70.3% top-1 throughput ratios respectively. At the same time, in order to ensure a 95% expected throughput ratio our model needs to sweep on average 6 beams, greatly reducing the beam search overhead. As a comparison, the two baselines requires 28 and 11 beams, respectively. In terms of NLOS performance, our model outperfoms both alternatives providing an average top-1 throughput ratio of 79.0%, improving upon the two baselines by 25% and 4%, respectively. Finally, our proposed model achieves a tighter 95% confidence interval in comparison with [16]–[18] specifically for smaller k. This is very favourable in practice as it ensures more reliable performance guarantees for different instances of training and deployment of our model.

V. Conclusions

We have proposed a supervised learning scheme for efficient mmWave beam selection that exploits side information in the form of LIDAR and position data to reduce the beam search overhead. Our approach significantly outperformed the state of the art in terms of beam classification accuracy and resulting throughput. We have introduced a non-local attention block to improve the performance, specifically for the more challenging NLOS scenarios. Additionally, we have proposed a curriculum learning strategy and a novel loss function inspired by knowledge distillation, which improved the training speed and the accuracy of the final solution. Our NN-based beam selection scheme was able to harness almost 79.9% of the available throughput without any beam search, just by utilizing the LIDAR side information. Moreover, we have showed that our NN architecture can be further simplified by pruning up to 60% of its parameters without considerable performance loss, hence, rendering it applicable for memory and computationally constrained applications.

References

[1] J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, "Millimeter-wave vehicular communication to support massive automotive sensing," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, 2016.

[2] V. Va, J. Choi, and R. W. Heath, "The impact of beamwidth on temporal channel variation in vehicular channels and its implications," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5014–5029, 2016.

[3] L. Wei, Q. Li, and G. Wu, "Initial access techniques for 5G NR: Omni/beam SYNC and RACH designs," in *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018, pp. 249–253.

[4] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. Thomas, and A. Ghosh, "Multilevel millimeter wave beamforming for wireless backhaul," in *2011 Ieee Globecom Workshops*. IEEE, 2011, pp. 253–257.

[5] V. Va, J. Choi, T. Shimizu, G. Bansal, and R. W. Heath, "Inverse multipath fingerprinting for millimeter wave V2I beam alignment," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4042–4058, 2018.

[6] M. Brambilla, M. Nicoli, S. Savaresi, and U. Spagnolini, "Inertial sensor aided mmWave beam tracking to support cooperative autonomous driving," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[7] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "Beam alignment for millimeter wave links with motion prediction of autonomous vehicles," in *Antennas, Propagation RF Technology for Transport and Autonomous Platforms 2017*, 2017, pp. 1–8.

[8] N. González-Prelcic, R. Méndez-Rial, and R. W. Heath, "Radar aided beam alignment in mmWave V2I communications supporting antenna diversity," in *2016 Information Theory and Applications Workshop (ITA)*, 2016, pp. 1–7.

[9] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, "Steering with eyes closed: mm-Wave beam steering without in-band measurement," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2416–2424.

[10] A. Ali, N. González-Prelcic, and R. W. Heath, "Millimeter wave beam-selection using out-of-band spatial information," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 1038–1052, 2018.

[11] M. Alrabeiah and A. Alkhateeb, "Deep learning for mmWave beam and blockage prediction using sub-6 GHz channels," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5504–5518, 2020.

[12] I. Chafaa, R. Negrel, E. V. Belmega, and M. Debbah, "Federated channel-beam mapping: from sub-6ghz to mmwave," in *IEEE WCNC 2021 Workshop: Distributed Machine Learning*, 2021.

[13] V. M. De Pinho, M. L. R. De Campos, L. U. Garcia, and D. Popescu, "Vision-aided radio: User identity match in radio and video domains using machine learning," *IEEE Access*, vol. 8, pp. 209 619–209 629, 2020.

[14] M. Alrabeiah, A. Hredzak, and A. Alkhateeb, "Millimeter wave base stations with cameras: Vision-aided beam and blockage prediction," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[15] W. Xu, F. Gao, S. Jin, and A. Alkhateeb, "3D scene-based beam selection for mmWave communications," *IEEE Wireless Communications Letters*, vol. 9, no. 11, pp. 1850–1854, 2020.

[16] A. Klautau, N. González-Prelcic, and R. W. Heath, "LIDAR data for deep learning-based mmWave beam-selection," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 909–912, 2019.

[17] M. Dias, A. Klautau, N. González-Prelcic, and R. W. Heath, "Position and LIDAR-aided mmWave beam selection using deep learning," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2019, pp. 1–5.

[18] M. B. Mashhadi, M. Jankowski, T.-Y. Tung, S. Kobus, and D. Gunduz, "Federated mmWave beam selection utilizing LIDAR data," *arXiv:2102.02802v1*, 2021, arXiv:2102.02802v1.

[19] *AI/ML in 5G*, accessed April 2021, https://www.itu.int/en/ITU-T/AI/challenge/2020/Pages/default.aspx.

[20] *AI/ML in 5G*, accessed April 2021, https://www.itu.int/en/ITU-T/AI/challenge/2020/Pages/results.aspx.

[21] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *NIPS 2014 Deep Learning Workshop.*, 2014.

[22] *Raymobtime*, accessed November 2020, https://www.lasse.ufpa.br/raymobtime/.

[23] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, "5G MIMO data for machine learning: application to beam-selection using deep learning," in *2018 Information Theory and Applications Workshop (ITA)*, 2018, pp. 1–9.

[24] P. I. Yonglong Tian, Dilip Krishnan, "Contrastive representation distillation," in *Proceedings of the International Conference on Learning Representations, ICLR 2020, April 2020, Addis Ababa, Ethiopia*.

[25] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[26] G. Hacohen and D. Weinshall, "On the power of curriculum learning in training deep networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*.

[27] A. Buades, B. Coll, and J. . Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, 2005, pp. 60–65 vol. 2.

[28] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[29] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems*, 1990, pp. 598–605.

[30] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.