AirNet: Neural Network Transmission over the Air

Mikolaj Jankowski, Deniz Gündüz and Krystian Mikolajczyk
Dept. of Electrical and Electronic Engineering, Imperial College London, UK
{mikolaj.jankowski17, d.gunduz, k.mikolajczyk}@imperial.ac.uk

***Abstract*—State-of-the-art performance for many emerging edge applications is achieved by deep neural networks (DNNs). Often, the employed DNNs are location- and time-dependent, and the parameters of a specific DNN must be delivered from an edge server to the edge device rapidly and efficiently to carry out time-sensitive inference tasks. This can be considered as a joint source-channel coding (JSCC) problem, in which the goal is not to recover the DNN coefficients with the minimal distortion, but in a manner that provides the highest accuracy in the downstream task. For this purpose we introduce AirNet, a novel training and analog transmission method to deliver DNNs over the air. We first train the DNN with noise injection to counter the wireless channel noise. We also employ pruning to identify the most significant DNN parameters that can be delivered within the available channel bandwidth, knowledge distillation, and non-linear bandwidth expansion to provide better error protection for the most important network parameters. We show that AirNet achieves significantly higher test accuracy compared to the separation-based alternative, and exhibits graceful degradation with channel quality.**

***Index Terms*—Neural network compression, joint source-channel coding, network pruning, distributed inference**

I. Introduction

An increasing number of edge devices are capable of carrying out complex signal processing and inference tasks. Currently, the state-of-the-art performance for many emerging edge applications is achieved by deep neural networks (DNNs). It is normally assumed that a DNN trained for a specific task is stored on the edge devices, e.g., an autonomous car, a drone or a mobile phone, to carry out inference on collected data. However, with the growing adoption of data-driven machine learning technologies, it will not be possible to store the parameters of all DNNs that may be needed by a device. Moreover, a DNN may be specific to a location or may be updated frequently due to non-stationarity of the environment, and it may need to be acquired by the edge device at the time of inference.

In this work, we consider scenarios, in which the parameters of a DNN have to be transmitted from an edge server (e.g., a base station), which has access to training data, to an edge device (e.g., an autonomous car) over a wireless channel for a time-sensitive inference task, as shown in Fig. 1. The conventional approach would be to first train a DNN, which is then compressed to be delivered efficiently over the bandwidth-limited channel. This approach can benefit from the existing literature on DNN training and compression.

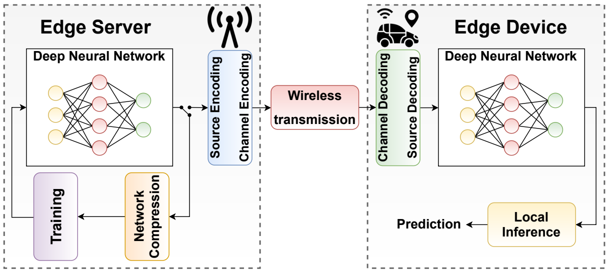


Fig. 1: System model.

We propose an alternative "analog" strategy for the bandwidth-efficient delivery of DNN parameters over a wireless channel. We utilize a novel joint source-channel coding (JSCC) approach, which directly maps the DNN parameters to channel symbols in an analog manner. Our strategy, called AirNet, allows us to greatly reduce the bandwidth as well as the computational burden of encoding and decoding. Our approach also reduces the requirement for accurate channel estimation. To the best of our knowledge, this is the first work that considers wireless transmission of DNN parameters for rapid edge inference applications. Our specific contributions can be summarized as follows:

- We propose a novel wireless DNN training and transmission scheme under bandwidth and transmission power constraints. In addition to employing network pruning and knowledge distillation to compress the DNN dimensions during training, we show that the non-linear Shannon-Kotelnikov (SK) mapping can provide unequal error protection and further bandwidth efficiency.
- We show that the proposed AirNet scheme achieves a satisfactory level of accuracy, while significantly reducing the bandwidth compared to state-of-the-art digital schemes, which employ DNN compression followed by channel coding.
- We perform extensive evaluations of our scheme, considering different channel models, training strategies, and channel conditions, and show that it consistently outperforms digital approaches.

II. Related Work

Although DNNs provide significant performance improvements for many different tasks, they usually require high computational and memory resources. From the communications perspective, the memory footprint of a DNN is a crucial factor. On the other hand, it is known that DNNs are usually

This work was supported in part by the European Research Council (ERC) through project BEACON (No. 677854) and UK EPSRC grant EP/N007743/1 under the CHIST-ERA program.

overparameterized, and their size can be reduced by compressing or removing (pruning) some redundant parameters [1]. Many different pruning techniques have been proposed [2]–[4]. In [2], Taylor expansion is utilized to approximate the change in the loss function induced by pruning to decide which parameters to remove, [3] considers l_1-norm of the network weights, while [4] studies the statistical information from a layer to prune the previous layer. Authors of [5] study the information-theoretic basis of pruning, assuming independent and identically distributed (i.i.d.) DNN weights with exponential distribution. Quantization can also be used to compress DNN parameters (please see [6] for a survey). In [7], quantized DNN parameters are further compressed by utilizing context-adaptive binary arithmetic coding, minimizing the impact of compression on the overall performance of the network. Another approach is to design compact and computationally efficient DNN architectures [8], rather than first training a large network and pruning it.

As we have highlighted above, the considered DNN delivery problem is a JSCC problem. Although Shannon's separation theorem [9], dictates the optimality of separate source and channel coding, it holds under the assumptions of infinite source and channel bandwidths, ergodic source and channel distributions, and an additive distortion measure in general, all of which are violated in our problem.

Many systems have been shown to benefit from designing the source/channel codes jointly. More recently, DNN-based efficient JSCC techniques have been shown to outperform conventional digital approaches even for the wireless transmission of well-studied sources such as images [10]–[13], speech [14], or videos [15]. Deep JSCC has also been applied to other downstream tasks, such as remote inference problems [16], [17], retrieval [18], or anomaly detection [19] problems. In our work, the goal of the receiver is to reconstruct a DNN, but we measure its quality by the accuracy in the desired inference task. Moreover, unlike images, DNN parameters may not necessarily follow a common statistics that can be exploited for efficient compression or JSCC; however, when training data is available, a particular DNN architecture can be trained or fine-tuned specifically for efficient wireless delivery. The similar problem of wireless delivery of DNN parameters is studied in [20], but in the absence of training data.

III. Methods

A. System model

We consider an edge server, with a large database of training samples. We assume that edge devices connect to this server to download the parameters of the model to perform inference on their local data samples. Our goal is to ensure the best possible inference performance, under power and bandwidth constraints on the channel from the edge server to the edge devices. Please see Fig. 1 for an illustration of our model.

The channel between the encoder and the decoder is modelled as a complex slow fading channel $\mathbf{y} = h\mathbf{x} + \mathbf{z}$, where $\mathbf{x} \in \mathbb{C}^b$ and $\mathbf{y} \in \mathbb{C}^b$ are the channel input and output vectors, respectively, $\mathbf{z} \in \mathbb{C}^b$ is the independent zero-mean unit-variance complex Gaussian noise vector, and $h \in \mathbb{C}$ is the complex channel gain. Here, b represents the available channel bandwidth limited due to the delay constraint of the downstream task. We assume that the channel gain h remains the same throughout the transmission, but changes from one transmission to the next in an i.i.d. fashion. An average power constraint is imposed on the channel input, that is, the channel input vector must satisfy $\frac{1}{b}\sum_{i=1}^{b}|x_i|^2 \leq P = 1$.

We will first consider the static additive white Gaussian noise (AWGN) channel, by setting $\|h\|$ to be a constant with a zero imaginary counterpart. We will also consider different channel conditions by varying channel's average SNR, defined as $\mathrm{SNR} = 10\log_{10}(\mathbb{E}[\|h\|^2])$ in the dB scale.

For the fading channel experiments, we assume that the channel state information (CSI) is available at the receiver. Therefore, the received signal $\mathbf{y} = h\mathbf{x} + \mathbf{z}$ is first multiplied by h^*, which is the complex conjugate of h and divided by its squared norm $\|h\|^2$. The resulting signal $\mathbf{x} + \frac{h^*\mathbf{z}}{\|h\|^2}$ is equivalent to an AWGN channel with a time-varying SNR.

At the receiver, we assume that the received symbols $\mathbf{y} \in \mathbb{C}^b$ are decoded into the parameters of a d-dimensional DNN, $\tilde{\mathbf{w}} \in \mathbb{R}^d$, and used for obtaining local predictions $p = g\left(a \mid \tilde{\mathbf{w}}\right)$, where a is an input, and $g\left(\cdot \mid \tilde{\mathbf{w}}\right)$ is a function representing the neural network's forward pass. The goal is to achieve the maximal possible accuracy with this inference task.

B. Training strategy

In the proposed delivery scheme, the trained neural network weights will be delivered over the wireless channel in an analog fashion; that is, they will be mapped directly to the channel inputs rather than being first compressed into bits, which are then coded and mapped to discrete constellation points. This has two consequences: First, we need to train the DNN so that it can be delivered using b channel symbols. Second, the receiver will recover noisy network weights whose values will depend on the channel realization. In order to achieve satisfactory network accuracy, the DNN must be trained in a way that guarantees robustness to channel imperfections. Our training strategy consists of the following steps.

The encoder first trains a d-dimensional DNN with parameters $\mathbf{w} \in \mathbb{R}^d$, where $d \gg b$, which will then be pruned to the available channel bandwidth. Choosing a large DNN as an initial point, rather than directly training a DNN of dimension b is motivated by recent findings in the pruning literature [21], which show that pruning a large DNN is generally easier than finding low-dimensional sub-DNNs that would achieve the same accuracy as the large DNN after being trained.

At each training iteration, we inject a certain amount of noise to the network's weights, as we hypothesize that the network can learn robustness against channel noise if it experiences it during training. The details about our noise injection strategy can be found in Section III-D.

Note that, with this scheme, each network parameter is transmitted as a single channel symbol, and cannot benefit from coding. However, it is known that some network parameters are more important than others for the inference task, and we may want to protect those better against channel noise. Therefore, we also consider pruning the network to less than b parameters, and then expanding it with a non-linear bandwidth expansion method, as described in Section III-F.

C. Network pruning

To reduce the network dimension to the channel bandwidth, we adopt a simple pruning method [3], where 10% of the remaining convolutional filters with the smallest l_1-norm are removed from the network at each step. In order to re-gain the accuracy, after each pruning iteration the network is fine-tuned. During fine-tuning we utilize both noise injection and knowledge distillation to ensure satisfactory performance of the network under noisy conditions. In this work, we assume that the side-information about the pruned DNN's structure (the number of filters remaining in each layer) is reliably transmitted to the receiver as metadata.

D. Noise injection

Noise injection has been originally proposed as a regularization technique to prevent overfitting in DNNs [22]. We note that in our setting such a strategy not only prevents overfitting, but also allows the network to adapt to the noisy channel characteristics for efficient inference. Therefore, at each training iteration, we calculate the network's predictions as $p = g\left(a \mid \tilde{\mathbf{w}}\right)$, where $\tilde{\mathbf{w}} = \eta(\mathbf{w})$ is a noisy set of network's weights, and $\eta(\cdot)$ represents either an AWGN or a fading channel, as described in Section III-A.

E. Knowledge distillation

Knowledge distillation [23] has been shown to be an effective method for increasing performance of small DNN models. The main idea behind knowledge distillation is to transfer some knowledge from a large DNN model, called the *teacher*, to a smaller model, denoted as the *student*. The loss function in knowledge distillation is defined as:

$$L_{total} = -t^2 \sum_i \hat{p}_i \log p_i - \sum_i \bar{p}_i \log p_i \tag{1}$$

where $\hat{p}_i = e^{\frac{\tilde{p}_i}{t}} / \sum_j e^{\frac{\tilde{p}_j}{t}}$ are the soft softmax predictions of the teacher model, t is the temperature parameter, which we set to 2 in all our experiments, $\bar{p}_i$ are the ground truth predictions, and p_i are the student network's predictions.

F. SK expansion for analog error correction

SK mapping is a method for performing efficient bandwidth compression or expansion in analog transmission [24]. The main idea is to project source symbols onto a lower- or higher-dimensional space, in order to reduce the bandwidth or counter the channel noise by expanding the bandwidth. In our problem, we first prune the network to a dimension smaller than b, and then expand it back to dimension b. The motivation for doing this, rather than directly pruning to dimension b, follows from the fact that DNNs can be pruned quite aggressively without significant loss in performance, while the SK expansion allows unequal error protection across DNN parameters.

SK mapping has been discussed in [25] for JSCC. The authors use Archimedes' spiral for encoding, and show its usefulness in the analog compression and expansion tasks. In our work, we employ a similar approach for bandwidth expansion using Archimedes' spirals, defined as:

$$x_1 = \frac{\Delta}{\pi} w \cos(w),\ x_2 = \frac{\Delta}{\pi} w \sin(w),\ w > 0, \tag{2}$$

$$x_1 = -\frac{\Delta}{\pi} w \cos(-w + \pi),\ x_2 = -\frac{\Delta}{\pi} w \sin(-w + \pi),\ w < 0, \tag{3}$$

where Δ is a scaling factor, which we fix to 0.1. We map each network parameter w to a point (x_1, x_2) on the 2D space. We encode the sign of the parameters by assigning positive-valued parameters to the spiral parameterized by (2), and the negative-valued parameters to the spiral parameterized by (3). At the receiver, we map the 2D points back to the original values by:

$$\hat{w} = \pm \underset{w}{\operatorname{argmin}} \left((x_1 - \theta(w))^2 + (x_2 - \theta(w))^2 \right), \tag{4}$$

where $\theta(\cdot)$ represents the union of the spirals defined in Eqs. (2) and (3), and the sign depends on which spiral the decoded point belongs to. Using the above formulas on all the DNN parameters allows us to perform $1:2$ bandwidth expansion. In order to achieve higher orders of expansion, we simply map the resulting points (x_1, x_2) from the 2D space to a higher dimensional space with the same mappings.

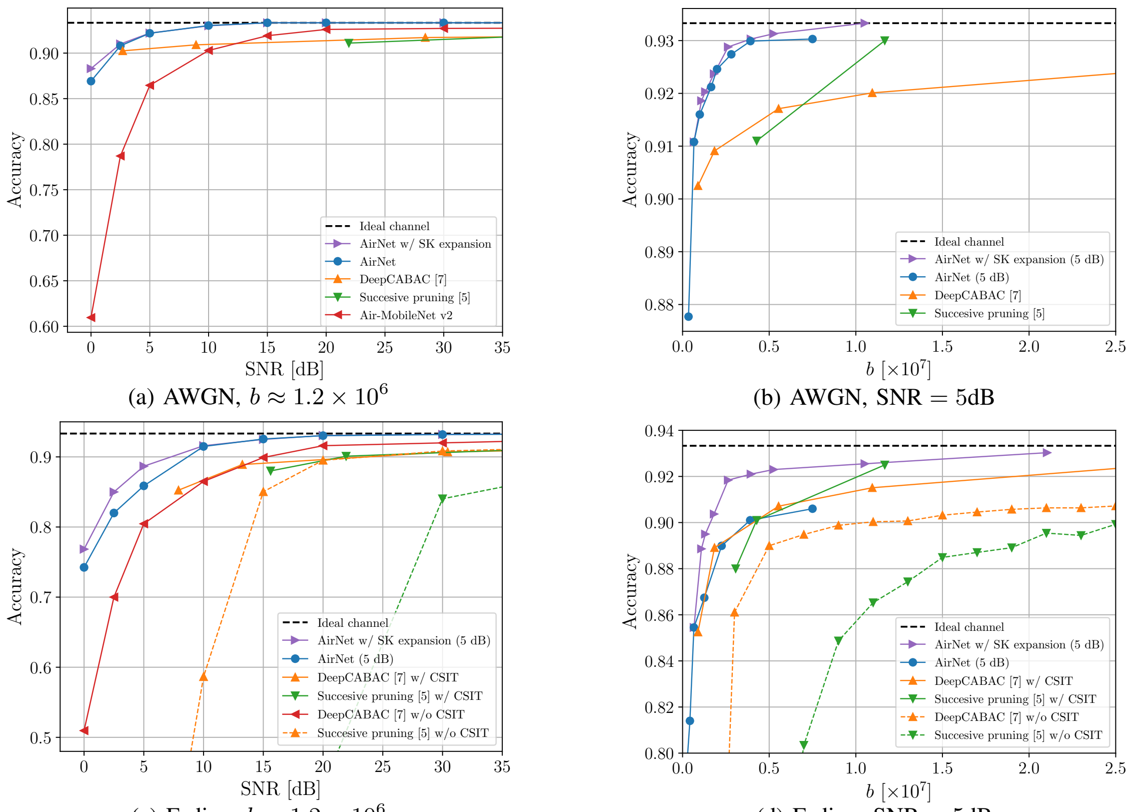
We note that the aforementioned solution only allows us to achieve expansion ratios of $1 : 2^n$, where n is the number of successive expansion steps. In order to achieve intermediate levels of expansion, we propose a simple algorithm for selecting a subset of layers to be expanded, instead of expanding the entire network uniformly. In the algorithm, we first calculate the predictions of the original network, without noise. Subsequently, we perform iterative evaluations, where at each iteration we inject noise to only one layer at a time, and calculate the mean squared error (MSE) between the original predictions and the predictions produced by the network with one of the layers perturbed with noise. Finally, we perform SK expansion of the layers, which, after perturbation, result in the highest MSE, thus are the most sensitive to noise.

IV. Results

In this section we evaluate the performance of the proposed AirNet and compare it with other schemes in the literature.

A. Experimental setup

In this work, we consider image classification task over CIFAR10 [26] dataset which contains 60000 RGB images of size 32×32 pixels, divided into 10 different classes. For a fair comparison with [5], [7], we consider Small-VGG16 [27], which employs the same convolutional layers as standard VGG16, but utilizes a different classifier head, which consists of two linear layers, first containing 512 neurons, followed by ReLU activation, and the second containing 10 neurons for class predictions. We perform multiple training runs of our network, for different values of training SNR, channel types,



(a) AWGN, $b \approx 1.2 \times 10^6$

(b) AWGN, SNR $= 5$dB

(c) Fading, $b \approx 1.2 \times 10^6$

(d) Fading, SNR $= 5$dB

Fig. 2: Performance comparison between AirNet, digital, and analog schemes over AWGN and slow fading channels.

pruning ratios (depending on the channel bandwidth b) and different training strategies. For knowledge distillation, we use ResNet-50 [28], trained on CIFAR10 dataset, as the teacher. For DNN training, we use SGD optimizer with learning rate of 0.01 and momentum of 0.9 for 30 epochs, reduce the learning rate to 0.001, and train for further 30 epochs.

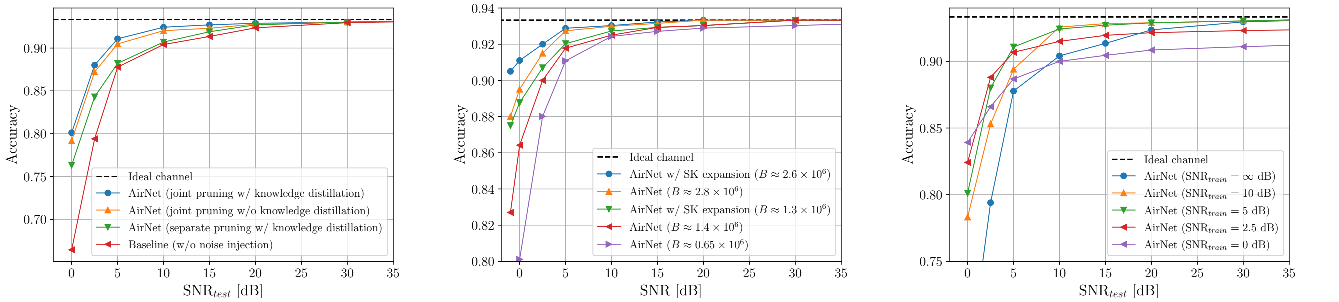
We compare our method to two state-of-the-art digital DNN compression approaches - DeepCABAC [7], and succesive pruning [5]. Both methods first perform DNN sparsification with pruning, quantize the remaining DNN parameters, and encode them into a minimal-length bitstream with arithmetic or Huffman coding, respectively.

For the analog schemes, we employ the channel model described in Section III-A with receiver CSI only. For the digital schemes, we consider two alternatives. When the CSI is available only at the receiver, the transmitter transmits at a fixed rate. If the channel capacity is below this rate, the transmission is considered to be failed. Then, we calculate the fraction of successful transmissions and calculate the resulting mean performance of transmitted DNNs. The second scenario assumes that the CSI is available also at the transmitter. In this case, the transmitter compresses the DNN to the rate dictated by the channel capacity. Please note that in both scenarios we consider digital transmission at the Shannon capacity, which is a rather generous upper bound on the performance.

B. *Inference performance*

In this section, we compare AirNet with alternative digital methods [5], [7] at different channel SNRs and bandwidths. As an additional baseline, we introduce Air-MobileNet v2, which has the same structure as popular MobileNet v2 [8], but is trained similarly to our models. We consider both AWGN and fading channels. Please note that the analog methods are trained with noise injection corresponding to the channel model used for testing.

The performance comparison for an AWGN channel with fixed bandwidth, is shown in Fig. 2a. Our method achieves satisfactory performance even for low channel SNRs. The digital alternatives tend to require a much higher SNR to allow successful transmission of the DNN parameters. Our network is able to recover almost perfect accuracy when the channel SNR values are above 10dB, whereas digital approaches require SNR > 50dB to achieve the same level of accuracy. Air-MobileNet v2 achieves satisfactory performance at high SNR regime, but fails when SNR is below 10dB. This behaviour is probably caused by the noise sensitivity of certain operations utilized in its structure, e.g., inverted residuals or linear bottlenecks. For the fixed channel SNR (Fig. 2b), we observe that our network requires less bandwidth compared to the digital alternatives. It can be also observed that with SK expansion following initial pruning of 91% of the weights, even better performance is achieved, especially at low SNRs.



(a) Different training strategies, $b \approx 0.65 \times 10^6$, $\mathrm{SNR}_{train} = 5$dB

(b) Different bandwidth, $\mathrm{SNR}_{train} = 5$dB

(c) Different training SNR, $b \approx 0.65 \times 10^6$

Fig. 3: Accuracy vs. SNR for different hyperparameter selections in training AirNet. In (a) we fix training SNR and bandwidth, and vary the training strategy; in (b) the training SNR is fixed, while we change the available bandwidth; in (c) we fix the bandwidth and vary the SNR used for training. All the experiments are performed over an AWGN channel.

Similar behaviour is observed for fading channels (Fig. 2c), where AirNet outperforms the digital approaches, especially when they also do not have access to instantaneous CSI at the transmitter (CSIT). AirNet is able to perform well in the fading regime, where the channel gain can differ between transmissions. However, higher SNR of at least 20dB is required to recover noiseless accuracy. SK improves the performance in the low SNR regime. Results for a fixed SNR (Fig. 2d) indicate that the expansion is necessary to achieve satisfactory accuracy, as AirNet, even at high bandwidths, fails to recover the accuracy that is close to the noiseless bound.

C. Performance for different training strategies

In this section, we compare different training strategies for AirNet. The results are shown in Fig. 3a. We observe that each step presented in Section III-B is crucial for the performance of our network. The best accuracy, for a fixed bandwidth, is achieved when we combine all the methods together, namely pruning with noise injection (indicated as *joint pruning* in Fig. 3a), and knowledge distillation. We see that knowledge distillation from a larger model allows us to achieve a small gain in the accuracy. Another important factor is to combine pruning with noise injection. The network, which was first pruned, and then fine-tuned with noise injection (denoted as *separate pruning*), achieves weaker performance, only slightly higher from the network trained without noise injection.

D. Performance for different bandwidths

The comparison between different bandwidths for AirNet is shown in Fig. 3b. We observe that as we increase the bandwidth, the robustness of the network against noise increases. For the AWGN case, bandwidth of roughly 5.2×10^6 channel symbols is sufficient to achieve the accuracy of 90% even at $\mathrm{SNR} = 0$dB. However, as we further reduce the bandwidth, we sacrifice the robustness. Another finding is that the SK expansion scheme is able to recover the accuracy loss due to pruning the network. In other words, it is better to first prune the network to a very low bandwidth and then expand it with SK mapping, compared to pruning to a moderate bandwidth. The advantage of the SK expansion becomes even more crucial at $\mathrm{SNR} < 5$dB as it provides better protection for the more significant network parameters.

E. Graceful degradation

In Fig. 3c, we present the performance of networks trained at different SNR_{train} values, and tested on a wide spectrum of SNR_{test} values. We note that AirNet, trained at a moderate SNR, achieves satisfactory performance even with a relatively large mismatch between the training and test SNRs; while the accuracy is maximized when the two match. Networks trained for low SNR_{train} fail to recover the full accuracy even when the channel improves. Again, we see that the network trained without noise injection ($\mathrm{SNR}_{train} = \infty$) performs the worst when the channel is noisy. We also observe that AirNet exhibits *graceful degradation*; that is, its performance slowly degrades as the channel gets worse. On the contrary, digital transmission exhibits a *threshold behaviour*, where the accuracy sharply drops when the channel conditions are worse than the code rate.

V. Conclusions

We presented AirNet - a novel training and analog transmission strategy for rapid and efficient wireless delivery of inference capabilities, in particular, in the form of DNN parameters, without resorting to the conventional source and channel coding steps. The strategy consists of joint pruning and noise injection, which leads to low bandwidth requirements and high robustness against channel noise. We also applied knowledge distillation step to boost the performance. SK mapping for bandwidth expansion is proposed as an unequal error protection scheme to increase the robustness of the more critical network parameters against channel impairments. Our strategy consistently outperforms digital network compression methods for AWGN and fading channel scenarios, showing its promise for time-sensitive location-dependent edge inference applications in future networks.

References

[1] Y. LeCun, J. Denker, and S. Solla, "Optimal Brain Damage," in *Advances in Neural Information Processing Systems*, 1990.

[2] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning Convolutional Neural Networks for Resource Efficient Inference," in *International Conference on Learning Representations (ICLR)*, 2016.

[3] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. Graf, "Pruning Filters for Efficient Vonvnets," in *Int'l Conf. on Learning Repr. (ICLR)*, 2017.

[4] J.-H. Luo, J. Wu, and W. Lin, "ThiNet: A Filter Level Pruning Method for Deep Neural Network Compression," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[5] B. Isik, A. No, and T. Weissman, "Successive Pruning for Model Compression via Rate Distortion Theory," *arXiv:2102.08329*, 2021.

[6] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *arXiv:1710.09282*, 2020.

[7] S. Wiedemann et al., "DeepCABAC: A Universal Compression Algorithm for Deep Neural Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 4, pp. 700–714, 2020.

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted Residuals and Linear Bottlenecks," in *IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR)*, 2018.

[9] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[10] E. Bourtsoulatze, D. Burth Kurka, and D. Gündüz, "Deep Joint Source-Channel Coding for Wireless Image Transmission," *IEEE Trans. on Cognitive Comms. and Networking*, vol. 5, no. 3, pp. 567–579, 2019.

[11] D. B. Kurka and D. Gündüz, "Deepjscc-f: Deep joint source-channel coding of images with feedback," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 178–193, 2020.

[12] D. B. Kurka and D. Gunduz, "Bandwidth-agile image transmission with deep joint source-channel coding," *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 8081–8095, 2021.

[13] M. Yang, C. Bian, and H.-S. Kim, "Deep joint source channel coding for wireless image transmission with ofdm," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[14] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2434–2444, 2021.

[15] T.-Y. Tung and D. Gündüz, "Deepwive: Deep-learning-aided wireless video transmission," *arXiv:2111.13034*, 2021.

[16] J. Shao and J. Zhang, "Bottlenet++: An end-to-end approach for feature compression in device-edge co-inference systems," *arXiv preprint arXiv:1910.14315*, 2019.

[17] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Joint Device-Edge inference over wireless links with pruning," in *IEEE International Workshop on Signal Proc. Advances in Wireless Comm. (SPAWC)*, May 2020.

[18] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Wireless Image Retrieval at the Edge," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 89–100, 2021.

[19] A. E. Kalør, D. Michelsanti, F. Chiariotti, Z.-H. Tan, and P. Popovski, "Remote anomaly detection in industry 4.0 using resource-constrained devices," in *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2021, pp. 251–255.

[20] Y. Shao, S. C. Liew, and D. Gündüz, "Denoising noisy neural networks: A bayesian approach with compensation," *arXiv:2105.10699*, 2021.

[21] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations (ICLR)*, 2019.

[22] R. Zur, Y. Jiang, L. Pesce, and K. Drukker, "Noise Injection for Training Artificial Neural Networks: A Comparison With Weight Decay and Early Stopping," *Medical physics*, vol. 36, pp. 4810–8, 10 2009.

[23] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learn. and Rep. Learning Wrkshp.*, 2015.

[24] V. A. Kotelnikov, *The Theory of Optimum Noise Immunity*, McGraw-Hill Book Company, Inc., 1959.

[25] F. Hekland, P. A. Floor, and T. A. Ramstad, "Shannon-Kotelnikov Mappings in Joint Source-Channel Coding," *IEEE Transactions on Communications*, vol. 57, no. 1, pp. 94–105, 2009.

[26] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009.

[27] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556*, 2014.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.