Joint Source-Channel Coding for Wireless Image Transmission: A Neural Architecture Search Approach

Yuchen Yang and Shaowei Wang
School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China
Email: yuchenyang@smail.nju.edu.cn, wangsw@nju.edu.cn

***Abstract*—In this paper, we propose a joint source-channel coding (JSCC) scheme for wireless image transmission, where the encoder and decoder of the transmission system are implemented by convolutional neural networks. A neural architecture search (NAS) method is introduced to find promising network structures aiming at minimizing the distortion of the target image. Experimental results demonstrate that our proposed scheme, referred to as NAS-JSCC, significantly outperforms the conventional manually designed ones in terms of peak-signal-to-noise ratio in nearly all tested signal-to-noise ratio regions.**

***Index Terms*—Joint source-channel coding, mobile edge computing, neural architecture search, wireless image transmission.**

I. Introduction

Shannon's source-channel *separation theorem* is the cornerstone of the separate design of source coding and channel coding in modern communication systems, where the source coding tries to find the most efficient representation of the source information, while the channel coding fights channel noise by adding redundancy to improve the system reliability. However, the increasing demand on the system latency and robustness is posing a challenge to the underlying assumptions of the theorem: *(i)* infinite code length; *(ii)* stationary, ergodic source and channel. Violation of either could break down the optimality of the separated coding scheme [1], [2]. In addition, the performance of the separately coded system deteriorates severely once the channel conditions are below a threshold, displaying *cliff effect* [3] which damages the robustness of the desired system. Therefore, it is imperative to develop methods of joint source and channel coding (JSCC) to further improve the system performance.

In recent years, deep leaning has found its applications in numerous areas of wireless communications such as resource allocation [4], channel prediction [5] and signal processing [6]. JSCC leveraging the data-driven approaches not only achieves improved performance, but also exhibits considerable potential to integrate with mobile edge computing, making it highly valuable for future mobile networks. Typically, the encoder and decoder can be parameterized as deep neural networks, whose structures vary with different tasks. In [7], a recurrent neural network based JSCC is proposed for text transmission, which outperforms traditional separate coding scheme in terms of word error rate owing to the ability of preserving semantic information against noisy channels. Convolutional neural network (CNN) is applied to a JSCC system for wireless image transmission [3], which surpasses the classical separate coding baseline in low signal-to-noise ratio (SNR) regions, alleviating the cliff effect to a great extent. Inception module and skip-connection are introduced to the JSCC network in [8], which achieves performance gain in high SNR regions. In [9], feedback information is exploited by using multiple parallel encoders and decoders, which largely enhanced the image quality at the receiver side. The problem of adaptive-bandwidth transmission is explored in [10], where the proposed JSCC scheme enables receivers to choose only the subsets of the transmission bandwidth for image reconstruction, improving the system performance in bandwidth constrained situations. Furthermore, orthogonal frequency-division multiplexing is combined with JSCC networks to combat multi-path fading [11]. The proposed system outperforms traditional separately coded schemes with better transmission quality. However, complexity is induced due to the incorporation with explicit signal processing techniques.

The key bottleneck of the aforementioned methods lies in the design of JSCC networks, which depends heavily on human expertise and often lead to an error-prone process. To tackle this issue, neural architecture search (NAS) attracted much attention due to its capability to algorithmically search for optimal network architectures [12]–[15]. In [14], the architecture search is handled with reinforcement learning, where a recurrent neural network is developed as a controller to generate the hyperparameters of a feedforward CNN. The controller is trained with policy gradient algorithm to maximize the performance of the generated structure. In [15], a strategy to continuously relax the discrete architecture space is proposed to allow the architecture search to be implemented using gradient descent. The method achieves state-of-the-art performance with orders of magnitude faster search speed. However, the building block of the searched network, i.e. *the cell*, involves complex connection and induced additional complexity [16], making it less suitable for its deployment on mobile devices.

In this paper, we propose a novel JSCC scheme for wireless image transmission, where differentiable neural architecture search is employed to find promising architectures of JSCC

This work was supported in part by the National Natural Science Foundation of China under Grants 61931023 and U1936202.

ICC 2023 - IEEE International Conference on Communications | 978-1-5386-7462-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICC45041.2023.10279680

networks. The method fully exploits the continuous relaxation of the discrete layer-wise search space considered in our model, and enables efficient search by using gradient based architecture search strategy. Experimental results show that our proposed NAS-JSCC network significantly outperforms the manually designed networks and the classical separated coding schemes in terms of peak-signal-to-noise ratio (PSNR).

The rest of the paper is organized as follows. In Section II, the system model and the problem formulation are described. The proposed method is illustrated in detail in Section III. Experiments in different scenarios are implemented and analyzed in Section IV. In Section V, we conclude the paper.

II. System Model and Problem Formulation

Consider the problem of wireless image transmission over a noisy channel. The input image with 3 color channels of size $H \times W$ is denoted by a vector $\boldsymbol{x} \in \mathbb{R}^n$, where $n = 3 \times H \times W$. In each transmission, the encoder directly maps the input image to the complex signal $\boldsymbol{y} \in \mathbb{C}^k$ with the encoding function $\Psi_{\theta(a)} : \mathbb{R}^n \mapsto \mathbb{C}^k$, which is implemented by using a CNN with architecture a and parameter θ. Normalization is performed at the end of the encoder to ensure the output signal $\boldsymbol{z} \in \mathbb{C}^k$ to have an average power of P, i.e. $\frac{1}{k}\mathbb{E}(|\boldsymbol{z}|^2) = P$. Consequently, we have:

$$\boldsymbol{z} = \sqrt{\frac{kP}{\boldsymbol{y}^H \boldsymbol{y}}} \cdot \boldsymbol{y} \ , \tag{1}$$

where $(\cdot)^H$ denotes the Hermitian transpose. The normalized signal is then transmitted through the noisy channel $\mathbf{c}_\sigma : \mathbb{C}^k \mapsto \mathbb{C}^k$, after which the decoder receives the distorted signal and reconstruct the input image using a decoder network $\Phi_{\phi(a)} : \mathbb{C}^k \mapsto \mathbb{R}^n$ with architecture a and parameter ϕ. The ratio k/n is defined as the bandwidth compression ratio, indicating the degree to which the original image is compressed. For AWGN channel, the channel function is expressed as: $\boldsymbol{c}_\sigma(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{CN}(0, \sigma^2 \boldsymbol{I}_{k\times k})$ with σ^2 being the average power of the noise.

Following the above description, the image after a single transmission process can be expressed as:

$$\hat{\boldsymbol{x}} = \Phi_{\phi(a)} \circ \boldsymbol{c}_\sigma \circ \Psi_{\theta(a)}(\boldsymbol{x}) \ , \tag{2}$$

where $\circ$ implies function composition. $\hat{\boldsymbol{x}}$ should approximate the original image with the least possible distortion $d(\boldsymbol{x}, \hat{\boldsymbol{x}})$. For a determined network architecture a, this can be achieved by jointly optimizing the parameters of the encoder and decoder to reach the minimal average distortion:

$$(\theta(a)^*, \phi(a)^*) = \underset{\theta(a),\phi(a)}{\arg\min} \, \mathbb{E}_{p(\boldsymbol{x},\hat{\boldsymbol{x}})}(d(\boldsymbol{x}, \hat{\boldsymbol{x}})) \ . \tag{3}$$

In this paper, we adopt the squared l_2-norm as the distortion measure between the original image and the transmitted one, which is computed as: $d(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2$.

Notice that Eqs. (2)-(3) are computable only if the architecture a is determined from a search space. In this work, we employ the layer-wise search space as proposed in [16], where the l-th layer in the network can select the kernel size from set $\mathcal{K}_l$ and the filter number from set $\mathcal{F}_l$ to conduct the convolutional operation. We explicitly define the search space of kernel size and filter number as follows:

$$\begin{aligned} &\mathcal{K}_l = \{(3\times 3), (5\times 5), (7\times 7)\}, \quad l \in \{1, 2, .., L\} \\ &\mathcal{F}_l = \begin{cases} \{16, 32, 48\} & \text{if } l \in \{1, 2, ..., L\} \setminus \{L_e, L\} \\ \{c_{out}\} & \text{if } l = L_e \\ \{3\} & \text{if } l = L \end{cases} , \end{aligned} \tag{4}$$

where L_e denotes the last layer of the encoder, whose filter number is set to be c_{out} to ensure a fixed bandwidth compression ratio, and L denotes the last layer of the JSCC network, whose filter number is set to be 3 to guarantee 3 color channels of the transmitted image.

The l-th layer-wise search space $\mathcal{A}_l$ in the network is thereby represented by the Cartesian product of the two sets: $\mathcal{A}_l = \mathcal{K}_l \times \mathcal{F}_l$, and the overall search space of the network is further expressed as:

$$\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times ... \times \mathcal{A}_L \ . \tag{5}$$

The architecture a is determined if and only if the operation in each layer is determined. Compared with global or cell based search space, the layer-wise search space enables diverse operation search across all layers while avoiding the extra latency induced by the complex and fragmented connection between layers, which achieves both high flexibility and low computational complexity.

Given a search space $\mathcal{A}$, our objective is to find the optimal architecture $a \in \mathcal{A}$ such that when the parameters of the JSCC network are successfully trained on the training set $\mathcal{Y}$ according to Eq.(3), the average distortion over the validation set $\mathcal{X}$ is minimized. Consequently, the architecture search focused in this paper can be formulated as the following optimization problem:

$$\begin{aligned} \min_{a\in\mathcal{A}} \quad & \mathbb{E}_{p(\boldsymbol{x},\hat{\boldsymbol{x}})}(\|\boldsymbol{x} - \Phi_{\phi(a)^*} \circ \boldsymbol{c}_\sigma \circ \Psi_{\theta(a)^*}(\boldsymbol{x})\|^2) \\ \text{s.t.} \quad & C_1 : (\theta(a)^*, \phi(a)^*) = \underset{\theta(a),\phi(a)}{\arg\min} \, \mathbb{E}_{p(\boldsymbol{y},\hat{\boldsymbol{y}})}(\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2) \\ & C_2 : \quad \hat{\boldsymbol{y}} = \Phi_{\phi(a)} \circ \boldsymbol{c}_\sigma \circ \Psi_{\theta(a)}(\boldsymbol{y}) \\ & C_3 : \quad \boldsymbol{y} \in \mathcal{Y} \\ & C_4 : \quad \boldsymbol{x} \in \mathcal{X} \ , \end{aligned} \tag{6}$$

where constraint C_1, C_2 and C_3 mean that for any selected architecture a, the parameter of the network is optimal with respect to training set image. Constraint C_4 indicates that the objective function is evaluated using the validation set. In this work, the optimization is conducted for 10-layer JSCC networks with $L_e = 5$, where the first two layers of the encoder and the last two layers of the decoder are $2\times$-downsampling and $2\times$-upsampling convolutional layers, respectively.

III. Proposed method

The optimization task in Eq. (6) is an intractable combinatorial problem with approximately 4×10^8 potential architectures in the search space, evaluating even one of them would entail training the network from scratch. To handle the formulated

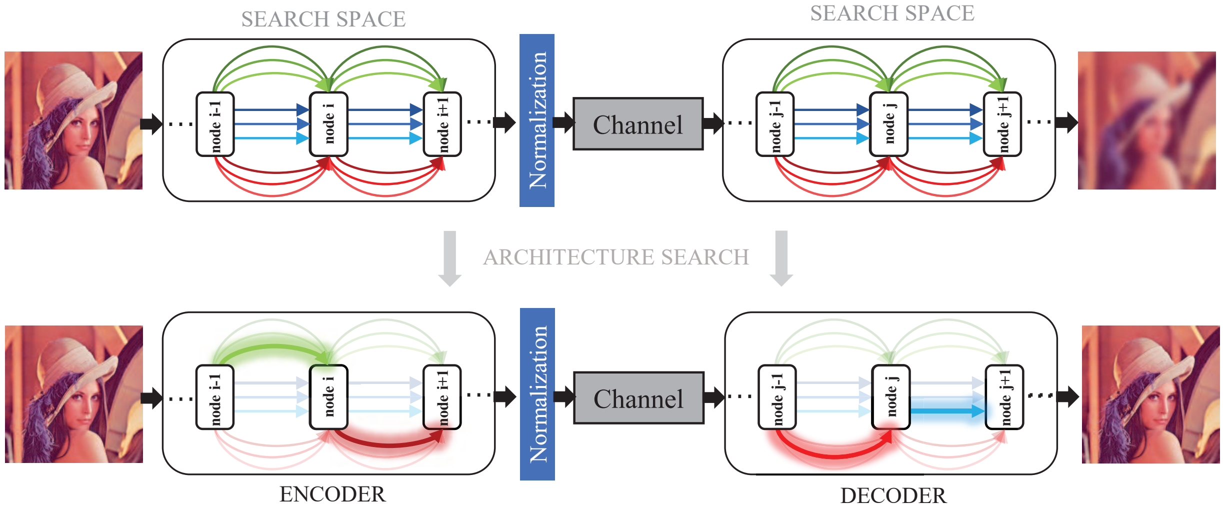

Fig. 1. Schematic diagram of the search procedure

problem, we employ the differentiable neural architecture search method. The key idea is to relax the discrete search space into a continuous one and optimize the architecture using gradient based strategy.

A. Continuous Relaxation of the Search Space

In the discrete search space, the candidate operations in each layer are viewed as directed edges between adjacent nodes. As depicted in Fig. 1, the network architecture is determined if and only if an exact operation is selected in each layer, thereby forming a path between the input node and the output node. The continuous relaxation is implemented by weighting and summing all the candidate operations in the l-th layer, where the weights are introduced as a set of continuous and trainable parameters, denoted by the vector $\boldsymbol{\alpha}^{(l)}$, which is normalized using Softmax function:

$$\alpha_i^{(l)} = \frac{e^{\gamma_i^{(l)}}}{\sum_{j=1}^{|\mathcal{A}_l|} e^{\gamma_j^{(l)}}} \ , \tag{7}$$

with l being the layer index, and $\boldsymbol{\gamma}^{(l)}$ the trainable parameters. Let $o_i^{(l)}(\cdot)$ be the i-th candidate operation in the l-th layer-wise search space $\mathcal{A}_l$. Consequently, the output of a single layer can be expressed as the weighted sum, i.e. the expectation of these operations:

$$\overline{o}^{(l)}(\boldsymbol{x}) = \sum_{i=1}^{|\mathcal{A}_l|} \frac{e^{\gamma_i^{(l)}}}{\sum_{j=1}^{|\mathcal{A}_l|} e^{\gamma_j^{(l)}}} \cdot o_i^{(l)}(\boldsymbol{x}) \ . \tag{8}$$

In this way, the discrete search space is relaxed to the continuous and trainable probabilities of the distribution over the candidate operations, and the architecture index a becomes the set of trainable operation weights $\alpha = \left\{\boldsymbol{\alpha}^{(l)}\right\}$. Different from the original search space where the information flows through a specific path between the input and output nodes, in the relaxed space, the information flows through all the paths between the input and the output node simultaneously, with the weights representing the likelihood of each candidate operation. The procedure of searching for the optimal architecture is thereby transformed into optimizing the distribution of the candidate operations. When the search procedure converges, we need to discretize the search space to perform architecture selection. As illustrated in the lower part of Fig. 1, we use the $argmax$ strategy to heuristically choose the most weighted operations in each layer to form the overall network architecture. Note that when there exists a weight that converges to 1 in every layer of the network, the continuously relaxed architecture converges to a specific architecture in the original search space accordingly.

B. The Search Algorithm

To make the notations more compact, the transmission function in Eq. (2) can be rewritten as:

$$\hat{\boldsymbol{x}} = f_{w(\alpha)}(\boldsymbol{x}, \alpha) \ , \tag{9}$$

where $w(\alpha) = (\phi(\alpha), \theta(\alpha))$ is the parameter set of the encoder and decoder networks. We denote the average distortion over the training set and the validation set by $\mathcal{L}_{\mathcal{Y}}$ and $\mathcal{L}_{\mathcal{X}}$, respectively:

$$\begin{aligned} \mathcal{L}_{\mathcal{Y}}(w(\alpha), \alpha) &= \mathbb{E}_{p(\boldsymbol{y},\hat{\boldsymbol{y}})}(\left\|\boldsymbol{y} - f_{w(\alpha)}(\boldsymbol{y}, \alpha)\right\|^2), \quad \boldsymbol{y} \in \mathcal{Y} \\ \mathcal{L}_{\mathcal{X}}(w(\alpha), \alpha) &= \mathbb{E}_{p(\boldsymbol{x},\hat{\boldsymbol{x}})}(\left\|\boldsymbol{x} - f_{w(\alpha)}(\boldsymbol{x}, \alpha)\right\|^2), \quad \boldsymbol{x} \in \mathcal{X} \ . \end{aligned} \tag{10}$$

Hence, the continuously relaxed version of Eq. (6) can be formulated as follows:

$$\begin{aligned} &\underset{\alpha}{min} \quad \mathcal{L}_{\mathcal{X}}(w(\alpha)^*, \alpha) \\ &s.t. \ \ w(\alpha)^* = \arg\min_{w} \mathcal{L}_{\mathcal{Y}}(w(\alpha), \alpha) \ . \end{aligned} \tag{11}$$

Note that this is a bilevel optimization problem, where the upper-level variable α encodes the architectural information, and the lower-level variable w is the network parameter. For each α, the derivation of $w(\alpha)^*$ requires a full training process of the network, which is infeasible due to computation cost. To tackle this issue, the optimal parameter $w(\alpha)^*$ is approximated

Algorithm 1 Architecture search for NAS-JSCC

Input $\mathcal{A}$, $\mathcal{X}$, $\mathcal{Y}$, η_α, η_w, δ, $\mathbf{c}_{\boldsymbol{\sigma}}$

Initialize: $\boldsymbol{\gamma}^{(l)} = 0, l = 1, 2, ..., n$

Compute: $\alpha_i^{(l)} = \frac{e^{\gamma_i^{(l)}}}{\sum_{j=1}^{|\mathcal{A}_l|} e^{\gamma_j^{(l)}}}, i = 1, 2, ..., |\mathcal{A}_l|$

repeat

  Update: $\alpha \leftarrow \alpha - \eta_\alpha \nabla_\alpha \mathcal{L}_{\mathcal{X}}(w - \delta \nabla_w \mathcal{L}_{\mathcal{Y}}(w, \alpha), \alpha)$

  Update: $w \leftarrow w - \eta_w \nabla_w \mathcal{L}_{\mathcal{Y}}(w, \alpha)$

until α is converged

for $l = 1, 2, ..., n$ **do**

  $i \leftarrow \arg\max_j \alpha_j^{(l)}$

  $o^{(l)}(\cdot) = o_i^{(l)}(\cdot)$

end for

Output: The searched NAS-JSCC network structure

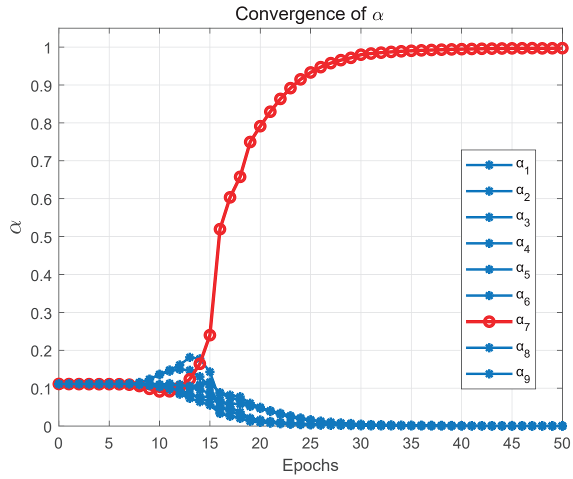

Fig. 2. Operation weights as a function of search epoch

by a single gradient update, i.e. $w(\alpha)^* \approx w - \delta \nabla_w \mathcal{L}_{\mathcal{Y}}$ [15]. The intuition behind the approximation is from related techniques in meta learning [17], where we aim to obtain the network parameters that are adaptable and transferable to other network architectures such that only a small number of gradient steps will lead to huge performance gain. In effect, the upper-level variable α is optimized to improve the overall performance of the architecture, and the lower-level variable w is trained to obtain high transferability, which in turn promotes the optimization of α. Thereby, the prohibitive training process of the inner parameters w is evaded, and the subsequent gradient steps of the architectural parameter α will converge efficiently.

We are now able to employ a gradient based strategy to solve Eq. (11), which is detailed as follows. First, we input the required search space, dataset, learning rates and a channel function. We consider AWGN channel in this work. We initialize the trainable parameters $\boldsymbol{\gamma}^{(l)}$ to be 0, which equalizes all of the operation weights in α. Second, the architecture parameters α is updated by a single gradient descent step using the aforementioned approximation. Third, the inner parameter w of the network is updated using gradient descent on the training set data. The second and the third step are repeated until the architecture α is converged. Then, we determine the overall architecture by heuristically selecting the most weighted operation in each layer, after which we train the searched NAS-JSCC network from scratch. The outline of the procedure is given in Algorithm 1.

IV. Experimental Results

We conduct our experiments on CIFAR-10 dataset, which contains 50000 training data and 10000 testing data. The training data is randomly splitted in half to obtain the validation data before searching. In the search phase, the learning rate η_α, η_w are set to 10^{-3} and 0.015, respectively. The step length δ is equal to η_w. Cosine annealing is applied to both the searching and the training phase. Adam optimization is used to update the network parameters, and the operation weights are updated using stochastic gradient descent. Moreover, nonlinearity is added by PReLU function in the first 9 layers, and by Sigmoid function in the last layer. We adopt the mean squared error function to approximate the average distortion in the objective functions of (6) and (11):

$$\mathcal{L} = \mathbb{E}_{p(\boldsymbol{x}, \hat{\boldsymbol{x}})}(\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2) \approx \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2 \ , \tag{12}$$

with N being the batch size, which is set to 128. The search is conducted in AWGN channel at a predefined SNR each time, with the transmission power P normalized to 1. The SNR is thus computed by:

$$\text{SNR} = 10 \cdot \log_{10}\left(\frac{1}{\sigma^2}\right) \ . \tag{13}$$

The performance of each network structure is evaluated in terms of PSNR.

Fig. 2 shows the convergence property of the architecture search. The JSCC network is searched when the channel SNR is equal to 15dB and the bandwidth compression ration is 1/6.

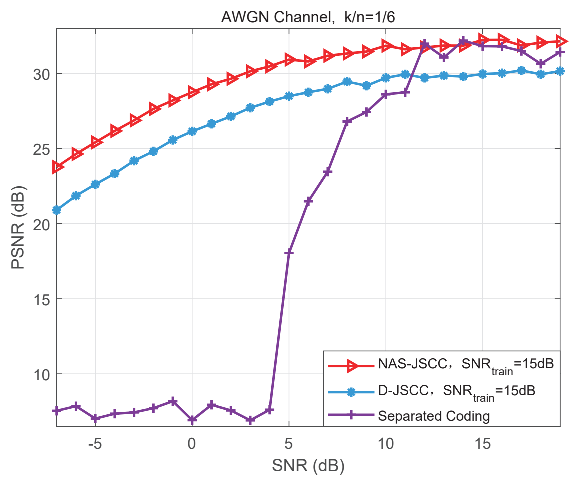

Fig. 3. PSNR performance as a function of test SNR

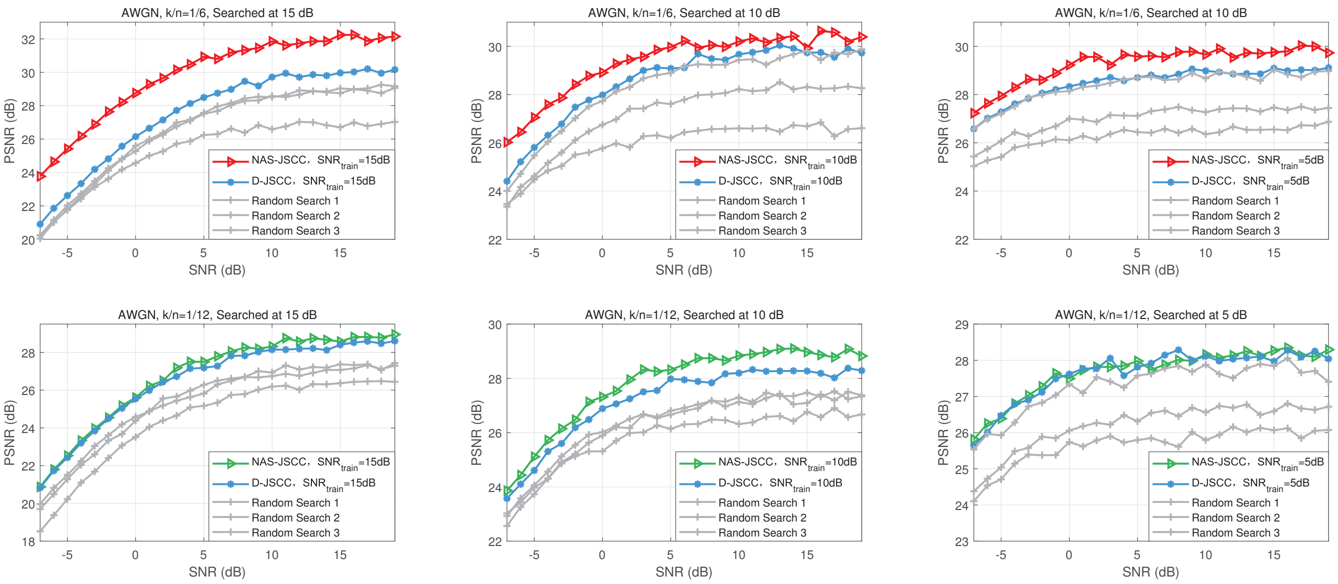

Fig. 4. PSNR performance as a function of test SNR

The weight of each operation in the first layer is examined during the first 50 epochs of iteration. The weights for other layers display the same behavior, and are not shown due to the length of the paper. At the beginning of the search, the 9 candidate operations have equal weights of $1/9$. The weight α_7 attenuates during the first 10 epochs of search, and is not the most weighted before the first 15 epochs. This reveals that the architecture during this phase changes with epoch and is not converged. After 15 epochs, the rest of the weights attenuate to 0 while α_7 rapidly converges to 1. Consequently, the operation corresponding to α_7 becomes the target operation to be selected at the end of the search.

Fig. 3 compares NAS-JSCC with a manually designed baseline, referred to as D-JSCC [3] and the separately coded scheme. In the separated coding scheme, we adopt the JPEG algorithm for source compression and the 1/2-rate low-density parity-check code for channel coding. Besides, the Quadrature Phase Shift Keying is also introduced for digital modulation. The NAS-JSCC is searched at 15dB, and both of NAS-JSCC and D-JSCC are trained at 15dB with compression ratio equal to 1/6. All evaluations all repeated 100 times for each test SNR to remove randomness. The separated coding scheme outperforms the manually designed baseline in high SNR regions. However, its performance deteriorates drastically when channel condition is below 10dB, displaying the notorious *"cliff effect"*. The NAS-JSCC shows superiority in all SNR regions, outperforming the D-JSCC and the separated coding significantly with relatively subtle performance degradation in poor channel conditions.

Fig. 4 compares the performance of the proposed NAS-JSCC networks with D-JSCC and the networks derived using random search. We conduct the search in 6 different scenarios where the channel SNR are 15dB, 10dB, 5dB and the compression ratio are 1/6 and 1/12, respectively. All networks have the same depth to avoid the unfairness brought by extra computation, and are trained with the same setting until convergence. The experimental results demonstrate that our proposed NAS-JSCC networks reconstruct the image with the best quality in nearly all scenarios, exhibiting a performance gain of nearly 2dB when searched and trained at the channel SNR of 15dB. Note that the random search method achieves the worst performance, indicating that randomness doesn't contribute to the performance gain and the gradient based strategy indeed generated satisfactory architectures from the search space.

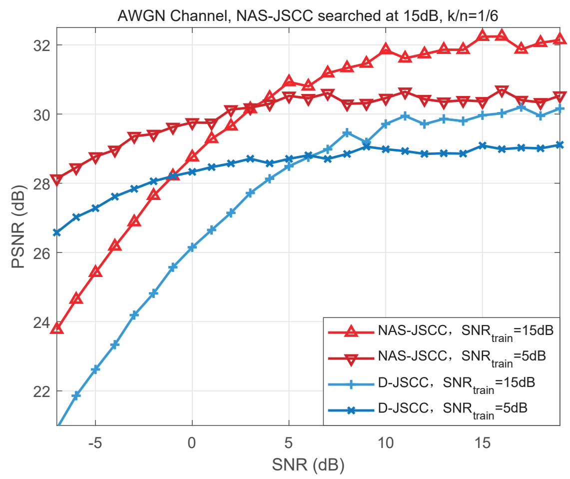

Fig. 5. PSNR performance as a function of test SNR in different training settings

We study the adaptability of NAS-JSCC when the SNR for searching mismatches the SNR for training in Fig. 5. We conduct the architecture search at the channel SNR of 15dB and train the searched network at 15dB and 5dB respectively. Results show that networks trained at higher SNR performs better at higher test SNR regions, while networks trained at lower SNR exhibit more gentle degradation when the channel condition deteriorates. Besides, NAS-JSCC searched at 15dB outperforms D-JSCC in all tested SNR regions when trained at 5dB, and in regions where SNR>0 when trained at 15dB, showing that the proposed scheme has significantly better performance. The results further demonstrate that the architecture search at a specific SNR does not affect the adaptability of the searched network, which allows the searched NAS-JSCC to be trained in various SNRs to adapt to different channel qualities.

V. Conclusion

In this paper, we presented a novel NAS-JSCC scheme for wireless image transmission. The differentiable neural architecture search method is employed to search for promising structures of the encoder and decoder network to minimize the distortion of the transmitted image. Experimental results demonstrate that the proposed NAS-JSCC network significantly outperforms the classical separated coding scheme and the manually designed JSCC network in various scenarios, which indicates a prosperous future of incorporating automatically designed networks into the task of image and video compression and transmission.

References

[1] S. Vembu, S. Verdu, and Y. Steinberg, "The source-channel separation theorem revisited," *IEEE Trans. Inf. Theory*, vol. 41, no. 1, pp. 44–54, Jan. 1995.

[2] V. Kostina and S. Verdu, "Lossy joint source-channel coding in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 2545–2575, Jan. 2013.

[3] E. Bourtsoulatze, D. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, Sept. 2019.

[4] W. Yu, T. Wang, and S. Wang, "Multi-label learning based antenna selection in massive MIMO systems," *IEEE Trans. on Vehic. Tech.*, vol. 70, no. 7, pp. 7255–7260, Jul. 2021.

[5] Z. Tao and S. Wang, "Improved downlink rates for FDD massive MIMO systems through Bayesian neural networks-based channel prediction," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 2122–2134, Mar. 2022.

[6] S. Liu, T. Wang, and S. Wang, "HIENet: A hardware impairment estimation network for OFDM systems," in *Proc. IEEE GLOBECOM'22*, Rio de Janeiro, Brazil, Dec. 2022.

[7] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *Proc. IEEE ICASSP'18*, Calgary, AB, Canada, Apr. 2018.

[8] X. Huang *et al.*, "A novel convolutional neural network architecture of deep joint source-channel coding for wireless image transmission," in *Proc. IEEE WCSP'21*, Changsha, China, Oct. 2021.

[9] D. Kurka and D. Gündüz, "DeepJSCC-f: Deep joint source-channel coding of images with feedback," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 178–193, Dec. 2020.

[10] D. Kurka and D. Gündüz, "Bandwidth-agile image transmission with deep joint source-channel coding," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 8081–8095, Dec. 2021.

[11] M. Yang, C. Bian, and H.-S. Kim, "OFDM-guided deep joint source channel coding for wireless multipath fading channels," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 584–599, Jun. 2022.

[12] T. Elsken, J. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1997–2017, Mar. 2019.

[13] P. Ren *et al.*, "A comprehensive survey of neural architecture search: Challenges and solutions," *ACM Comput. Surveys*, vol. 54, no. 4, pp. 1–34, May 2021.

[14] B. Zoph and Q. Le, "Neural architecture search with reinforcement learning," in *Proc. ICLR'17*, Toulon, France, Apr. 2017.

[15] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *Proc. ICLR'19*, New Orleans, LA, USA, May 2019.

[16] B. Wu *et al.*, "FBNet: Hardware-aware efficient ConvNet design via differentiable neural architecture search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019.

[17] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML'17*, Sydney, Australia, Aug. 2017.