Machine Learning-based Source-Matched Channel Coding for Speech Transmission

Oemer Karakas[1], Andreas Brendel[2], Marco Breiling[1], Guillaume Fuchs[2], Sahana Raghunandan[1], Wolfgang Gerstacker[3]

[1] Broadband and Broadcasting Department, Fraunhofer IIS, Erlangen, Germany

[2] Audio and Multimedia Engineering Department, Fraunhofer IIS, Erlangen, Germany

[3] Institute of Digital Communication, FAU Erlangen-Nürnberg, Erlangen, Germany

***Abstract*—This paper proposes a novel machine learning-based source-matched channel coding approach for transmission of short speech frames. We use a state-of-the-art speech codec to compare conventional channel coded transmission, uncoded transmission, and the proposed scheme. Our results demonstrate that our separate source and channel coding approach for short frames achieves superior performance compared to a state-of-the-art joint source and channel coding (JSCC) approach. By keeping separated source and channel coding, we take a step towards addressing network-related aspects, and we allow for independent training of the source codec and thus reduce the training complexity of the overall transmission system. Additionally, we present results on peak-to-average power ratio (PAPR) constrained transmission to facilitate the implementation of the proposed approach in real-world applications.**

***Index Terms*—Speech Transmission, Source Matched Channel Coding, Machine Learning-based Channel Coding.**

I. Introduction

The ultimate goal of communication is the exchange of information, that is *relevant to the receiving user*. Conventional communication systems employ separate and independent source and channel coding functions, where source coding aims at representing the source information in the most compact form possible into (binary) symbols / bits, whereas channel coding encodes these symbols - agnostic to the used source coding - for a reliable transmission over the per se unreliable communication channel. For source and channel codes of infinite block length and with the objective of *perfect* (i.e., error-free) reconstruction, Shannon proved the theoretical optimality of this separation. However, in practical communication systems and with short block lengths, the separation is sub-optimum. Therefore, joint source and channel coding (JSCC) has attracted considerable research interest recently, and this has been further fueled by the advent of deep learning (DL), facilitating the realization and optimization of communication systems according to perceptually motivated losses. Since mobile data traffic currently represents a major contribution to the global energy consumption and is expected to further grow exponentially, source dependent communication becomes increasingly important for more resource-efficient transmission [1].

Perceptual quality-oriented communication systems, also considered as a certain class of semantic communication systems, have been explored across various domains using DL-based JSCC approaches including video, image, text and speech transmission [2], [3]. Corresponding works have demonstrated significant performance gains, typically measured in terms of perceptual evaluation metrics over signal-to-noise ratio (SNR) of the transmit channel, when compared to conventional systems in their respective fields. Moreover, JSCC methods that are designed for perceptual objectives typically exhibit a graceful degradation in performance with deteriorating channel quality, which highlights the robustness of this approach under variable transmission conditions.

In this work, we focus on the transmission of speech signals by employing a pre-trained neural speech codec (NSC) [4] (cf. Section III-B2) as the source coding component. We include a *separate channel* coding scheme that is trained *independently* of but specifically for this source codec and is hence matched to it. Therefore, we refer to our approach as *source-matched* channel coding (SMCC). Hence, our proposed scheme bridges the gap between conventional JSCC with its significant gains and the complete separation of source and channel coding. This allows, e.g., to store source-encoded data (like audiobooks) on servers and transmit them on-demand to users using the source-matched channel codec, or to transmit a source-encoded payload via multiple hops (e.g., mobile phone to base station, backhaul link, and remote base station to remote phone) - each individually protected by SMCC while the source coding remains end-to-end.

Our results demonstrate that SMCC achieves superior performance compared to a selected state-of-the-art JSCC scheme. Furthermore, over a large range of received speech qualities, it outperforms conventional transmission schemes that rely on the same source codec, emphasizing the advantages of tailoring channel coding to the source codec. Our findings also reveal that the output bit rate of the source encoder can be reduced to levels similar to those in conventional systems. Finally, we demonstrate that our SMCC approach can operate under typical peak-to-average power ratio (PAPR) constraints, highlighting its practical feasibility for real-world applications.

II. System Model

In this section, we introduce a system model that can represent both the considered benchmark schemes and our SMCC approach. The system model comprises a transmitter,

a channel, and a receiver. The purpose of the transmission system is to convey speech at a constant frame duration.

A. Transmitter

A speech frame of duration T_{F}, i.e., a sequence of l_s equidistant samples of a speech record, is represented by vector $\boldsymbol{s} \in \mathbb{R}^{l_s}$. A source encoder maps $\boldsymbol{s}$ to a feature vector $\boldsymbol{f}$,

$$\boldsymbol{f} = \text{s\_enc}(\boldsymbol{s}) \in \mathbb{R}^{l_f}, \tag{1}$$

where $\text{s\_enc}(\cdot)$ and l_f denote the encoding operation and the length of feature vector, respectively. $\boldsymbol{f}$ comprises *binary* symbols in conventional schemes, however, we allow for *real-valued* symbols in order to include our SMCC approach as well. The transmitter block responsible for modulation and coding referred to as $\text{mod\_cod}(\cdot)$ accepts $\boldsymbol{f}$ as input and generates the real-valued encoded transmit symbol vector $\boldsymbol{a}$,

$$\boldsymbol{a} = \text{mod\_cod}(\boldsymbol{f}) \in \mathbb{R}^{l_a}, \tag{2}$$

where l_a denotes the number of transmit symbols per speech frame.

B. Channel

After continuous-time pulse shaping filtering, the continuous time channel, matched filtering and sampling, the received symbol sequence when represented by a vector $\boldsymbol{y}$ is given by,

$$\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{n} \in \mathbb{R}^{l_a}, \tag{3}$$

where $\boldsymbol{n} \in \mathbb{R}^{l_a}$ is the noise vector, whose elements are i.i.d Gaussian variables, $\boldsymbol{n} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, N_0/2)$. Here, $N_0/2$ represents the double-sided power spectral density of the white Gaussian noise in equivalent baseband.

C. Receiver

The block responsible for demodulation and channel decoding characterized by the function $\text{dem\_dec}(\cdot)$ accepts $\boldsymbol{y}$ as input and delivers an estimate of the feature vector,

$$\hat{\boldsymbol{f}} = \text{dem\_dec}(\boldsymbol{y}) \in \mathbb{R}^{l_f}. \tag{4}$$

Based on this, the source decoder $\text{s\_dec}(\cdot)$ produces an estimate of the input speech frame,

$$\hat{\boldsymbol{s}} = \text{s\_dec}(\hat{\boldsymbol{f}}) \in \mathbb{R}^{l_s}. \tag{5}$$

III. Reference Schemes

In this section, we describe the state-of-the-art channel and speech coding algorithms that serve as reference components as well as the benchmark transmission schemes for comparison with our proposed SMCC-based scheme, as detailed in Section V.

A. Channel Coding

Polar coding with a code rate of 1/3, successive cancellation list (SCL) decoding with list size of 64 is selected as a scheme representing conventional channel coding due to its favorable performance for short packets. Polar coding is implemented via the Nvidia Sionna library [5].

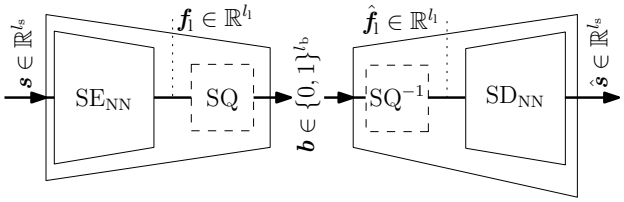

Fig. 1. NSC encoder and decoder block diagram.

B. Speech Coding

1) Enhanced Voice Services (EVS): We consider the 3GPP Enhanced Voice Services (EVS) standard [6] as a state-of-the-art non-neural speech coding approach. In our study, we operate EVS with 7.2 kbps and packet loss concealment.

2) Neural Speech Codec (NSC): The recently proposed real-time capable NSC [4] achieves low algorithmic latency and computational complexity, rendering it suitable for real-time communication applications. It employs a neural network (NN), $\mathrm{SE_{NN}}$, which encodes speech frames of length 20 ms sampled at 16 kHz ($l_{\mathrm{s}} = 320$) into a latent space representation $\boldsymbol{f}_{\mathrm{l}} \in \mathbb{R}^{l_{\mathrm{l}}}$ with $l_{\mathrm{l}} = 20$. The latent vectors are discretized using 3-bit scalar quantization (SQ) per dimension, producing a bit stream $\boldsymbol{b} \in \{0,1\}^{l_{\mathrm{b}}}$ with $l_{\mathrm{b}} = 60$. On the decoder side, a corresponding dequantization step is applied, followed by reconstruction through a decoding NN, $\mathrm{SD_{NN}}$. For further details, we refer to [4].

C. Transmission Benchmark Schemes

As baselines, we have chosen the following combinations of source and channel codecs:

1) EVS + Polar: Here, bits encoded by EVS with 7.2 kbps and Polar coding (see Section III A) are mapped to binary phase-shift keying (BPSK) symbols and decoded at the receiver side.

2) NSC + Polar: This scheme is similar to EVS + Polar, except that we adopt NSC with 3.0 kbps for the source coding.

3) NSC-L Uncoded: We use the $\mathrm{SE_{NN}}$ module of NSC for source coding. This transmission scheme does not use an explicit channel codec - instead, the latent representation extracted by the source encoder is first normalized to zero mean and unit variance (with mean and variance computed using the VCTK training set [7]) and then forwarded (as a non-channel coded real-valued sequence) to the transmit pulse shaping unit. After reception, the $\text{dem\_dec}(\cdot)$ module denormalizes the received signal in order to guarantee its original mean and variance. Speech is reconstructed using the resulting latent representation as input to the $\mathrm{SD_{NN}}$ module of NSC. With this scheme, we aim at exploring the graceful degradation property of the NSC source codec.

4) Deep Joint Source-Channel Analog Coding for Low-Latency Speech Transmission [3]: In this state-of-the-art neural JSCC scheme, the parameters of the NN are jointly optimized for both source and channel coding and specifically for perceptual speech quality.

IV. Proposed SMCC-based Scheme

In the following, we introduce the system model components of our proposed SMCC-based scheme and the corresponding training procedure. NNs are utilized for each functional block.

A. System Model Components

The NN block of NSC which maps the speech frame to a latent space representation, SE_{NN}, and the block regenerating the speech frame from the reconstructed latent vector, SD_{NN} (cf. Fig. 1), are used as source encoder s_enc$(\cdot)$ and source decoder s_dec$(\cdot)$, respectively. The source encoder and decoder networks are pre-trained and were not altered. Details regarding their architecture and training procedure are available in [4]. An NN, CE_{NN}, is employed in the mod_cod$(\cdot)$ unit and another NN, CD_{NN}, is utilized in the dem_dec$(\cdot)$ unit. In the sequel, we will refer to our proposed scheme as NSC+SMCC.

In contrast to conventional channel codecs, the NN responsible for channel encoding, CE_{NN}, accepts real-valued latent values as its input rather than *binary* symbols/bits. Its architecture is inspired from [8], and summarized in Table I. Each convolutional layer uses a kernel size of five, with zero-padding on both sides to maintain the input dimension. The stride, dilation and groups are chosen to one, i.e., each filter processes every input channel individually without skipping or dilating values, and each input channel is connected to every output channel. The first convolutional layer operates with a single channel, while the subsequent four layers have 50 input channels each. In all layers except the normalization layer a bias is applied, and the exponential linear unit (ELU) activation function is adopted. The dense layer reduces the output to a single channel, enabling the network to produce the vector that comprises the transmit symbols. We have considered two power normalization modes, both of which make sure that the long term average symbol energy is one. In the first mode, the system maintains a constant average transmit packet energy, while it enables the encoder to assign a higher packet energy to some latent representation vectors than for others. We refer to this mode as Dynamic Packet Energy Mode (DPEM). In the second mode, constant packet energy is enforced regardless of the content of the packet, denoted as Constant Packet Energy Mode (CPEM).

Additionally, we have evaluated a scheme with a soft delimiter operation at the last layer (which provides the transmit symbols) to account for a PAPR constraint.

The NN responsible for channel decoding, CD_{NN}, comprises 1D convolutional layers of the same type as CE_{NN}. The final layer is a dense layer with no activation. Its architecture is summarized in the lower part of Table I.

B. Training

CE_{NN} and CD_{NN} are jointly trained end-to-end. The training chain begins with the latent representation delivered by SE_{NN} as the input to CE_{NN} and ends with the latent representation estimate at the output of CD_{NN}. The training dataset consists of latent representations generated by SE_{NN} from speech recordings in the VCTK/training dataset, which is a well-known, widely adopted high-quality speech dataset. For each recording, latent representations are extracted for each speech frame of duration $T_{\text{F}} = 20$ ms, which are then stacked and shuffled to create the final training dataset, that is denoted by $\mathcal{D}$. The joint training of channel encoder and decoder minimizes a composite loss function defined as

$$\mathcal{L}_f(\boldsymbol{f}_{\text{l}},\hat{\boldsymbol{f}}_{\text{l}}) = (1-\lambda)\mathcal{L}_1(\boldsymbol{f}_{\text{l}},\hat{\boldsymbol{f}}_{\text{l}}) + \lambda\mathcal{L}_2(\boldsymbol{f}_{\text{l}},\hat{\boldsymbol{f}}_{\text{l}}), \tag{6}$$

where $\mathcal{L}_1(\cdot,\cdot)$ and $\mathcal{L}_2(\cdot,\cdot)$ denote the mean absolute error (MAE) and mean squared error (MSE), respectively. We empirically select $\lambda = 0.5$. We linearly combine MSE and MAE to avoid overemphasis on large errors. With E_p denoting the average transmit energy per transmit packet, i.e., per speech frame, the training process uses E_{p}/N_0 levels uniformly sampled (in dB domain) from the interval 12 dB $< E_{\text{p}}/N_0 <$ 20 dB, as values below 12 dB do not yield meaningful quality, while 20 dB already ensures near error-free transmission. The NN weights are updated using the Adam optimizer with a learning rate of 0.001, batch size of 3200, β values of 0.9 and 0.999, and no weight decay.

For the DPEM, batch normalization is applied during training to generate transmit symbols. After convergence, the weights and the normalization parameters, calculated over the entire speech records in the VCTK training set, are frozen and then used for inference. In contrast, in the CPEM, every packet is energy-normalized individually both in training and inference.

TABLE I
Architecture of CE_{NN} and CD_{NN}

Layer	Type	Input Chn	Output Chn	Activation
CE 1	Conv1D	1	50	ELU
CE $2-5$	Conv1D	50	50	ELU
CE 6	Dense	-	-	ELU
CE 7	Power Norm	-	-	None
CE 8	Clipping	-	-	None
CD 1	Conv1D	1	50	ELU
CD $2-8$	Conv1D	50	50	ELU
CD 9	Dense	-	-	None

V. Numerical Results and Discussion

This section presents our experiments, the used metrics, and a performance comparison.

A. Metrics

All records in the VCTK/Testing set are used to extract speech frames to be transmitted. NSC+SMCC and the first three benchmark schemes convey a 20 ms speech frame sampled at 16 kHz, while transmission benchmark scheme 4 transmits speech frames of $T_{\text{F}} = 8$ ms also sampled at 16 kHz. While we transmit one packet per speech frame, the different schemes yield different transmit packet lengths l_a due to the differences in the compression rate of the source coding and the modulation and coding rate of the channel coding scheme

TABLE II
TRANSMIT PACKET LENGTHS AND SYMBOL RATES FOR $T_{\mathrm{F}} = 20$ MS

Scheme	Src bits	Chn bits	l_a	R_{sym} (kHz)
EVS+Polar	144	432	432	21.6
NSC+Polar	60	180	180	9
NSC-L Uncoded	-	-	20	1
NSC+SMCC	-	-	20	1

(cf. Table II). Also the transmit symbol rates R_{sym} (rates of symbols transmitted over the channel) differ.

For performance assessment, we employ the Extended Short-Time Objective Intelligibility (ESTOI) Score, which is a well-known and broadly adopted objective speech quality measure with high correlation to the intelligibility of speech signals of varying quality. It ranges from 0 to 1, where a higher score means a better quality.

B. Discussion

Fig. 2 presents the perceptual quality (for error-free transmission) of the considered EVS and NSC speech codecs as horizontal lines. The ESTOI score is shown versus E_{p}/N_0 for a transmission over the AWGN channel for the first three benchmark schemes and NSC+SMCC, respectively. Here, an ESTOI score of approx. 0.7 corresponds to an acceptable quality and serves as our reference threshold. Comparing EVS + Polar and NSC + Polar, a reduction in source coding rate from 7.2 to 3 kbps achieves approximately a 3 dB improvement at the reference threshold. Surprisingly, NSC-L Uncoded outperforms NSC+Polar when the ESTOI score is 0.75 or lower. This improvement likely arises from two main factors: first, NSC employs bounded uniform noise during training to approximate quantization effects, which not only enhances quantization performance but also introduces inherent forward error correction capabilities in $\mathrm{SE_{NN}}$. Furthermore, uncoded transmission over AWGN benefits from graceful degradation.

At the reference threshold, NSC+SMCC shows a gain of approximately 3 dB over NSC-L Uncoded, approx. 5.2 dB over NSC+Polar, and roughly 8.2 dB over EVS+Polar, demonstrating the benefits of an SMCC approach, as the source codec is identical to that of NSC+Polar (except for the discarded quantization and dequantization modules). Conventional Shannon-based channel coding focuses on error-free transmission for SNRs above a target SNR, which manifests itself in the well-known "cliff effect" - here visible in the ESTOI performance of EVS+Polar and NSC+Polar. By contrast, NSC+SMCC attempts to provide best reconstruction of the latent representation for the complete *SNR range* used in the training, resulting in graceful degradation.

According to Table II, NSC+SMCC requires the least symbol rate $R_{\mathrm{sym}} = 1$ kHz and achieves thus the highest bandwidth efficiency. Increasing the number of transmit symbols and thus lowering the bandwidth efficiency results in general in a further improvement in power efficiency, i.e., an even lower required E_{p}/N_0 for a target speech quality. Similarly, if

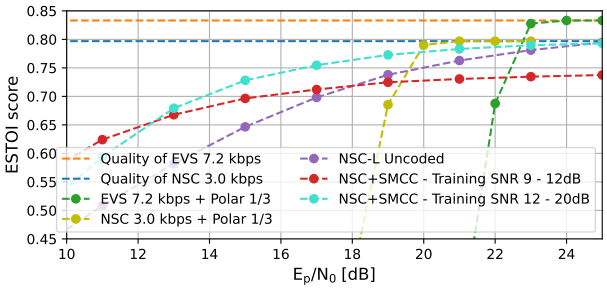

Fig. 2. ESTOI score vs. E_p/N_0 for different schemes.

we replace BPSK by higher-order modulation for EVS+Polar and NSC+Polar, the required E_{p}/N_0 will grow further.

The proposed channel encoder comprises 50.55k parameters and requires 1.01 MMacs (Mega Multiply-Accumulate Operations), while the corresponding channel decoder requires of 88.2k parameters and 1.76 MMacs. When executed on a single core of an Intel® Core™ i9-10900K CPU @ 3.70 GHz, encoding of a 20 ms audio frame takes 43 µs, and decoding requires 72 µs. In comparison, Simulating the Sionna implementation of polar coding under the given configurations takes more than 10 times longer. However, the measured CPU execution times may not fully reflect the actual efficiency due to the involvement of machine learning libraries. Notably, our proposed scheme employs cascaded convolutional layers, which are well-suited for acceleration on dedicated CNN hardware. Given their highly parallelizable nature, the channel encoder and decoder could theoretically be reduced to 8 and 17 clock cycles, respectively, under optimal hardware acceleration. The deep learning model underlying the used NSC [4] comprises 3.61 M parameters and 343 MMacs. Furthermore, it is implemented in a causal fashion with low delay which enables real-time communication (see [4] for details).

The widely-used Perceptual Evaluation of Speech Quality (PESQ) score does unfortunately not provide meaningful results for generative AI-based codecs like NSC, because they are not waveform-preserving. However, ESTOI scores are available for our fourth benchmark scheme from [3]. As its speech frame duration $T_{\mathrm{F}} = 8$ ms differs from the duration $T_{\mathrm{F}} = 20$ ms of the other considered schemes, we introduce the useful energy $E_{1\mathrm{ms}}$ received during the transmission of *one millisecond* of speech for a fair comparison: for NSC+SMCC we obtain $E_{1\mathrm{ms}}/N_0 = \frac{E_p}{20\ \mathrm{ms}} \cdot 1\ \mathrm{ms}/N_0$, whereas for the scheme in [3], we have $E_{1\mathrm{ms}}/N_0 = \frac{1}{2}\mathrm{SNR} \cdot \alpha \cdot n/8\ \mathrm{ms} \cdot 1\ \mathrm{ms}$ where the used SNR being valid for a real-valued transmission corresponds to $E_s/(N_0/2)$ (E_s: average transmit energy per symbol), n is the speech frame length (in samples) and $\alpha = l_a/n$ with transmit packet length l_a. The left part of Fig. 3 shows a comparison of NSC+SMCC and the scheme of [3] (curves for the parameter sets from [3] with $T_{\mathrm{F}} = 8$ ms, $n = 128$, SNR $= 0$ dB and 10 dB, $\beta = 100$, and varying α). Moreover, The left part of Fig. 3 shows the required symbol rates R_{sym} of the schemes.

It becomes apparent that for acceptable ESTOI scores above

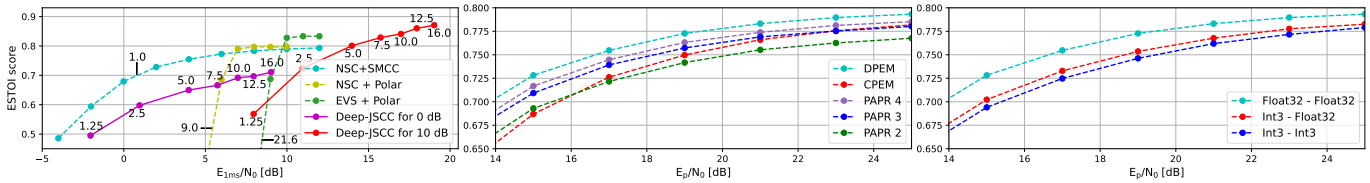

Fig. 3. (left) ESTOI score for NSC+SMCC and the best performing scheme of [3] (purple and red curve for varying α). The numbers along the curves represent the transmit symbol rates R_{sym} (in kHz). (middle) ESTOI score vs. E_p/N_0 of NSC+SMCC with PAPR constraints and (right) quantized source-channel codec interface.

0.7, our scheme achieves gains from 2.5 to 8 dB upon the JSCC scheme of [3], even though our scheme offers the additional advantage of separate source and channel coding, whereas the scheme of [3] uses NN parameters *jointly* optimized for source and channel coding. Moreover, the bandwidth efficiency of NSC+SMCC is higher.

In the middle part of Fig. 3, the performance of NSC+SMCC is shown under different transmit energy/power constraints. In particular, the impact of DPEM and CPEM is investigated. Additionally, the system performance is evaluated when constraining the PAPR to 4, 3, and 2 (in linear scale). A PAPR of 3 is particularly relevant as it corresponds to that for M-QAM/M-ASK constellations for $M \to \infty$. Here, a slight performance degradation of approximately 0.02 in ESTOI score results compared to the model without PAPR constraint.

Next we consider the effect of handing over 3-bit integer values (”Int3”) instead of 32-bit floating-point (”Float32”) at the interfaces between source and channel codecs. The right part of Fig. 3 shows performance differences of 0.02 to 0.03 in ESTOI score, where the left label (*Int3* or *Float32*) in the legend represents the interface between source and channel *encoder* and the right one that between channel and source *decoder*. Hence, the standard output of NSC (with 3 bits per latent representation) can be used with some performance penalty.

The observed gains over comparable schemes might be attributed to the absence of unequal-error-protection (UEP) in the Polar code and residual redundancy in the source-encoded latent vectors. Even though the employed NSC source codec is among the currently most efficient ones and hence supposed to reduce any redundancy as much as possible, we found by inspection of the encoder output that it still contains residual redundancy. This redundancy is reflected in statistical dependencies between the entries of the latent vector, and in a non-uniform probability distribution of the latent representations.

This paper does not include a JSCC scheme based on NSC because our focus is on the flexibility of training the proposed SMCC scheme independently of the NSC. This allows for rapid adaptation to different neural speech codecs without requiring full joint optimization. A JSCC scheme based on NSC will be explored in future work.

VI. Conclusion

We have found that it is possible to separate source and channel coding while realizing similar or even larger gains as with state-of-the-art joint source-channel coding (JSCC) approaches based on deep neural networks. Our channel codec is neural-network-based and optimized specifically for an input that is a latent space representation of speech and is produced by a state-of-the-art speech encoder - we refer to this as ”source-matched channel coding”. By focusing on channel coding optimization alone, the neural network training is simplified, as the source codec’s parameters remain fixed. This approach also allows source coding to remain on the application layer of the transmission network, such that source-encoded, i.e., compressed, data can be stored on servers instead of much larger raw source data. While the source coding and decoding is only carried out on both ends of the transmission chain, in a multi-hop transmission our channel coding can be applied at each hop’s physical layer. To enable compatibility with traditional communication protocols like TCP/IP and paradigms like end-to-end encryption, further research is needed to look into, e.g., source-matched encryption.

References

[1] W. Tong and G. Y. Li, “Nine challenges in artificial intelligence and wireless communications for 6G,” *IEEE Wireless Communications*, vol. 29, no. 4, pp. 140–145, 2022.

[2] N. Islam and S. Shin, “Deep learning in physical layer: Review on data driven end-to-end communication systems and their enabling semantic applications,” *IEEE Open Journal of the Communications Society*, vol. 5, p. 4207–4240, 2024.

[3] M. Bokaei, J. Jensen, S. Doclo, and J. Østergaard, “Deep joint source-channel analog coding for low-latency speech transmission over gaussian channels,” in *2023 31st European Signal Processing Conference (EUSIPCO)*, 2023, pp. 426–430.

[4] A. Brendel, N. Pia, K. Gupta, L. Behringer, G. Fuchs, and M. Multrus, “Neural speech coding for real-time communications using constant bitrate scalar quantization,” *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–15, 2024.

[5] J. Hoydis, S. Cammerer, F. A. Aoudia, A. Vem, N. Binder, G. Marcus, and A. Keller, “Sionna: An open-source library for next-generation physical layer research,” 2023. [Online]. Available: https://arxiv.org/abs/2203.11854

[6] M. Dietz, M. Multrus, V. Eksler, V. Malenovsky, E. Norvell, H. Pobloth *et al.*, “Overview of the evs codec architecture,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5698–5702.

[7] J. Yamagishi, C. Veaux, and K. MacDonald, “CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit,” 2019.

[8] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, “Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels,” 2019. [Online]. Available: https://arxiv.org/abs/1911.03038