Adaptive Semantic Communications: Overfitting the Source and Channel for Profit

Jincheng Dai, Member, IEEE, Sixian Wang, Graduate Student Member, IEEE, Ke Yang, Graduate Student Member, IEEE, Kailin Tan, Graduate Student Member, IEEE, Xiaoqi Qin, Member, IEEE, Zhongwei Si, Member, IEEE, Kai Niu, Member, IEEE, and Ping Zhang, Fellow, IEEE

Abstract—Most semantic communication systems leverage deep learning models to provide end-to-end transmission performance surpassing the established source and channel coding approaches. While, so far, research has mainly focused on architecture and model improvements, but such a model trained over a full dataset and ergodic channel responses is unlikely to be optimal for every test instance. Due to limitations on the model capacity and imperfect optimization and generalization, such learned models will be suboptimal especially when the testing data distribution or channel response is different from that in the training phase, as is likely to be the case in practice. To tackle this, in this paper, we propose a novel semantic communication paradigm by leveraging the deep learning model’s overfitting property. Our model can for instance be updated after deployment, which can further lead to substantial gains in terms of the transmission rate-distortion (RD) performance. This new system is named adaptive semantic communication (ASC). In our ASC system, the ingredients of wireless transmitted stream include both the semantic representations of source data and the adapted decoder model parameters. Specifically, we take the overfitting concept to the extreme, proposing a series of ingenious methods to adapt the semantic codec or representations to an individual data or channel state instance. The whole ASC system design is formulated as an optimization problem whose goal is to minimize the loss function that is a tripartite tradeoff among the data rate, model rate, and distortion terms. The experiments (including user study) verify the effectiveness and efficiency of our ASC system, the ingredients of wireless transmitted stream include both the semantic representations of source data and the adapted decoder model parameters. Notably, the substantial gain of our overfitted coding paradigm can catalyze semantic communication upgrading to a new era.

Index Terms—Semantic communications, online learning, data stream, model stream, rate-distortion tradeoff.

I. INTRODUCTION

Semantic communications are recently emerging as a new paradigm driving the in-depth fusion of information and communication technology (ICT) advances and artificial intelligence (AI) innovations [1]–[3]. Unlike traditional communication design philosophy that focuses on accurately transmitting bits over a noisy communication channel [4], semantic communications are goal-oriented, which helps the transceiver identify the most valuable information more efficiently, i.e., the information necessary to recover the purpose intended by the transmitter. Performance assessment also goes beyond the common Shannon paradigm of guaranteeing the correct reception of each single transmitted bit, human perceptual loss [5]–[7] and machine task accuracy [8] are taken as the distortion, which is better aligned with the essential goal of end-to-end communications.

The core idea of semantic communications is bridging the source and channel parts of Shannon theory together to boost the end-to-end transmission performance. The paradigm aiming at the integrated design of source and channel processing is joint source-channel coding (JSCC) [9], a classical topic in the information theory and coding theory. But conventional JSCC schemes [9]–[12] are based on explicit probabilistic models and handcrafted designs, whose optimization is intractable for complex sources. In addition, they ignore the semantic aspects of source messages and cannot be optimized towards human perception or machine task directly. Semantic communications leverage deep learning to realize JSCC [13]–[20], which can be well learned to optimize the end-to-end distortion. For wireless image transmission, these deep JSCC approaches have shown performance surpassing classical separation-based BPG source compression [21] combined with advanced low-density parity-check (LDPC) channel coding [22], especially for sources of small dimensions, e.g., tiny CIFAR10 image dataset. Recent semantic JSCC codecs [23], [24] were developed to optimize the reconstruction quality (distortion) with channel bandwidth cost (rate) jointly, which indeed reveals the core problem of semantic communication: transmission rate-distortion (RD) optimization. By introducing entropy model on the semantic latent representations, the coding efficiency of deep JSCC has been greatly improved [23], which well supports the transmission of large-scale sources. This coding paradigm perfectly describes the key aspects of semantic transmission for a large amount of images or videos: maximizing reconstruction quality meets the human perception; whereas minimizing wireless channel bandwidth cost benefits the efficient transmission.

Although existing optimized semantic communication approaches have proven to be very successful in minimizing the end-to-end expected RD cost over a full source dataset and ergodic wireless channel responses [23], they are yet unlikely to be optimal for every test instance due to the limited model capacity. Existing models only pay attention to an average low RD cost on the training set. For a given input data sample and wireless channel response, such a learned codec might not be good at capturing the data semantic feature and channel state at this instance, resulting in suboptimal transform and coding.

This work was supported in part by the National Natural Science Foundation of China under Grant 92067202, Grant 62001049, Grant 62071058, and Grant 61971062, in part by the Beijing Natural Science Foundation under Grant 4222012.

J. Dai, S. Wang, K. Yang, K. Tan, Z. Si, and K. Niu are with the Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: daijincheng@bupt.edu.cn).

K. Niu is also with Peng Cheng Laboratory, Shenzhen, China.

X. Qin and P. Zhang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China.
during the model inference stage. The imperfect optimization and generalization will be especially severe when the testing data distribution or channel response is different from that in the training phase. To tackle this, we explore a different approach by optimizing the semantic codec or representation individually, on a per data sample and channel response basis, during the model inference stage. In other words, we turn to minimize the end-to-end instant RD cost for substantial gains on every data and channel state instance.

Some insights in traditional source compression codec can bring some inspirations [25]–[27]. Conventional image/video compressors follow the hybrid transform coding paradigm. For example, HEVC [28] and VVC [29] jointly use discrete cosine transform (DCT) and discrete sine transform (DST) to handle different types of signals. Multiple transform selection (MTS) mechanism is introduced to VVC standard to select the most appropriate transform aiming at the best RD performance. Inspired by the idea of signal-dependent transform in traditional source compression methods, neural codecs are also evolving toward the instance-adaptive paradigm. The suboptimality of neural codecs has been studied extensively in terms of the model inference suboptimality [30]. It has been shown that by finetuning the encoder parameters or latent codes from a well-trained model for a particular instance, substantial gain can be obtained in the compression RD performance [31]–[36]. Some new methods are recently developed to further improve the RD cost, e.g., a full-model instance-adaptive compression method was proposed in [37], a neural syntax method was proposed in [38] to realize data-dependent compression, etc. All these methods leverage neural network’s overfitting property, adapting the model to an individual source data sample.

Inspired by the useful insights from traditional compression codecs and neural compression codecs, in this paper, we make the first attempt to build a neural instance-adaptive source and channel coding architecture for end-to-end semantic communications. We demonstrate that overfitted neural-enhancement on semantic communication systems is feasible and effective when combined with online learning. Accordingly, we propose adaptive semantic communication (ASC), a novel framework that employ online learning to overfit the instant source data sample and channel state information (CSI). Our new method is general and, as such, can be applied to a number of different architectures of semantic communications. In this paper, as a representative, we inject our online adaptation method to the nonlinear transform source-channel coding (NTSCC) based semantic communication system [23] to verify the effectiveness and efficiency of ASC. Unlike previous works, apart from the traditional content stream, our system additionally introduces adapted model parameters as the model stream to update the deep JSCC decoder and synthesis transform parameters at the receiver. We take into account the costs of sending the model updates such that the whole ASC system design is formulated as an optimization problem whose goal is to minimize the loss function that is a tripartite tradeoff among the data rate (R), model rate (M), and distortion (D) terms. Our proposed method is simple, effective and intuitively appealing, which could be feasibly transplanted to any existing deep learning based semantic communication systems. Fig. 1 illustrates the key idea of our work briefly.

Specifically, the contributions of this paper can be summarized as follows.

(1) ASC Framework: We make the first attempt to build a source and channel instance or domain adaptive semantic communication system based on NTSCC. Our innovative overfitting mechanism enables the whole NTSCC system to be much more powerful and flexible for extracting more compact semantic representations, offering superior end-to-end transmission RD performance. We make variational analysis to interpret the origins of general model inference suboptimality, and clarify the rationale of our proposed source and channel overfitting paradigm.

(2) Overfitting the Source: We design two simple yet efficient ways to realize the source data instance or domain adaptive NTSCC. These methods are of different ideas for adapting a given pre-trained model or semantic latent representation to an individual data sample or other content domain that is different in appearance. We discuss the specific scenario for which each method is suitable. Our source overfitting mechanism online adapts a baseline model to each specific scene to maximize communication efficiency.

(3) Overfitting the Channel: We propose a plugin-in CSI modulation module inserted into pre-trained codec modules. It enables the whole system to efficiently deal with different channel states with a single trained network. The proposed scheme can not only adapt to different signal-to-noise ratio (SNR) under the block fading channel, but also provide consistent and robust performance under diverse frequency selective fading channels, which makes our model agilely transferred over various channel states.

(4) Performance Validation: We verify the effectiveness and
efficiency of our ASC system over video I-frame sources and practical wireless channels. Results indicate that our method can lead to substantial gains in transmission RD performance. Equivalently, achieving the same end-to-end transmission performance (objective metrics, e.g. PSNR), our new ASC system can save up to 41% bandwidth cost compared to the state-of-the-art engineered transmission scheme (VVC combined with 5G LDPC coded transmission). A user study confirms that our approach is preferred to the state-of-the-art communication methods, even when they use more than double the wireless channel bandwidth cost. Some results discussions are further given to clarify the necessity of our overfitting paradigm for catalyzing semantic communications.

The remainder of this paper is organized as follows. In Section [I] we review the architecture and properties of NTSCC semantic communication system, and analyze its suboptimality using the variational inference. Next, in Section [III] we present our source overfitting methods, including different ideas for adapting a given pre-trained baseline model to an individual data sample or other content domain. In Section [V] we present our channel overfitting methods, including details on a plugin-in CSI modulation module to adapt a pre-trained model to the instant channel state. Section [VI] shows experimental results to quantify our performance gain, and some valuable discussions are also given. Finally, Section [VII] concludes this paper.

Notational Conventions: Throughout this paper, lowercase letters (e.g., $x$) denote scalars, bold lowercase letters (e.g., $\mathbf{x}$) denote vectors. In some cases, $x_i$ denotes the elements of $\mathbf{x}$, which may also represent a subvector of $\mathbf{x}$ as described in the context. Bold uppercase letters (e.g., $\mathbf{X}$) denote matrices, and $\mathbf{I}_m$ denotes an $m$-dimensional identity matrix. $\ln(\cdot)$ denotes the natural logarithm, and $\log(\cdot)$ denotes the logarithm to base 2. $p_x$ denotes a probability density function (pdf) with respect to the random variable $x$. In addition, $\mathbb{E}(\cdot)$ denotes the statistical expectation operation, and $\mathbb{R}$ denotes the real number set. Finally, $\mathcal{N}(x|\mu, \sigma^2) \triangleq (2\pi\sigma^2)^{-1/2} \exp(- (x-\mu)^2/(2\sigma^2))$ denotes a Gaussian function, and $\mathcal{U}(a-u, a+u)$ stands for a uniform distribution centered on $a$ with the range from $a-u$ to $a+u$.

II. PRELIMINARIES AND MOTIVATION

Built upon the variational auto-encoder (VAE) architecture, NTSCC has shown superior performance on wireless image and video transmission problems [23], [40]. Owing to the powerful ability of representation learning, NTSCC can well extract the source semantic features and transmit them over the wireless channels efficiently by using variable-length deep JSCC techniques. This method not only achieves comparable or better performance than state-of-the-art (SOTA) engineered source compression combined with advanced channel coding schemes, but also greatly surpasses plain auto-encoder based deep JSCC methods [13] that directly encode the raw source data rather than its semantic features. In addition, NTSCC shows great potential to achieve lower time complexity due to its efficient parallel computing with deep neural networks (DNNs). The above superiorities of NTSCC can well support semantic communications. Therefore, in this paper, we choose NTSCC to build the semantic communication system. Similar to previous works, we take image or video I-frame (intra-coded frame) source as representative, but our work is extensible for other source modalities.

A. NTSCC based Semantic Communication System

The idea of NTSCC stems from the landmark work of Ballé et al. on nonlinear transform coding (NTC) [41]–[43]. Given the pristine data sample $x$, e.g., an image $x$ modeled as a vector of pixel intensities $x \in \mathbb{R}^m$, it is first transformed into semantic latent representation $y$ using a DNN-based nonlinear analysis transform $g_a$. In data compression tasks, $y$ will be quantized as discrete-valued latent representation $\hat{y}$, followed by entropy encoding [44] to convert $\hat{y}$ into bit sequence. This bit sequence will be fed into entropy decoding to losslessly reconstruct $\hat{y}$, and another nonlinear synthesis transform DNN module $g_s$ uses $\hat{y}$ to reconstruct the decoded data $\hat{x}$. The $g_a$ and $g_s$ are jointly optimized under the RD constraint. In wireless communication systems, the above source compressive coding paradigm relies heavily on advanced channel coding and signal processing techniques to ensure the transmitted bit sequence to be losslessly recovered. This separation-based approach has been employed in many current communication systems, as the binary representations of various source data can be seamlessly transmitted over arbitrary wireless channels by changing the underlying channel code.

However, with increasing demands on low-latency wireless data delivery applications such as extended reality (XR), the limits of the separation-based design begin to emerge. Current wireless data transmission systems suffer from time-varying channel conditions, in which case the separation-based design leads to significant cliff-effect when the channel condition is below the level anticipated by the channel code [13]. Furthermore, the widely-used entropy coding is quite sensitive to the variational estimate of the marginal distribution of the source latent representation. Small perturbations on this marginal can lead to the catastrophic error propagation in entropy decoding [45]. In practice, the small perturbation is often caused by the floating point round-off error [46]. This round-off operation depends heavily on hardware and software platforms, and in various data compression applications, the transceiver may employ different platforms as stated in [46]. As a result, this non-determinism issue in transmitter vs. receiver will lead to severe performance degradation.

To address the above issues, our idea in NTSCC [23] is to replace quantization and entropy coding by integrating source coding and channel coding as a trained DNN, resulting in deep JSCC to transmit the latent representation $y$ directly. We have achieved better end-to-end transmission performance, and the system is robust to unpredictable wireless channels. The whole procedure of NTSCC is depicted in Fig. 2. The latent code $y$ is fed into both the analysis transform $h_a$ and the deep JSCC encoder $f_e$. On the one hand, $h_a$ summarizes the distribution of mean values and standard derivations of $y$ in the hyperprior $z$. The transmitter utilizes $z$ to estimate the mean vector $\mu$ and the standard derivation vector $\sigma$, and use them to determine the
bandwidth to transmit the latent representation. On the other hand, $f_c$ encodes $y$ as the channel-input sequence $s \in \mathbb{R}^k$, and the received sequence is $\hat{s} = W(s)$, whose transition probability is $p_{\hat{s}|s} (\hat{s}|s)$. In this paper, we consider the general fading channel model such that the transfer function is $\hat{s} = W(s|h) = h \odot s + n$ where $\odot$ is the element-wise product, $h$ denotes the CSI vector, and each component of the noise vector $n$ is independently sampled from a Gaussian distribution, i.e., $n \sim p_n \sim N(0, \sigma_n^2 I)$, where $\sigma_n^2$ is the noise power. At the receiver, $\hat{s}$ is further fed into the deep JSCC decoder $f_d$ to reconstruct the latent representation $\hat{y}$, which is further used by the nonlinear synthesis transform $g_s$ to recover the source data $\hat{x}$. The whole procedure of NTSCC system is

$$x \xrightarrow{\begin{smallmatrix} g_s(:, \phi_f) \end{smallmatrix}} y \xrightarrow{\begin{smallmatrix} f_s(:, \phi_f) \end{smallmatrix}} s \xrightarrow{W(\cdot| h)} \hat{s} \xrightarrow{\begin{smallmatrix} f_d(:, \theta_f) \end{smallmatrix}} \hat{y} \xrightarrow{\begin{smallmatrix} g_s(:, \theta_s) \end{smallmatrix}} \hat{x}$$

with the latent prior $y \xrightarrow{\begin{smallmatrix} h_s(:, \phi_h) \end{smallmatrix}} z \xrightarrow{\begin{smallmatrix} h_s(:, \theta_h) \end{smallmatrix}} \{\mu, \sigma\}$,

where $(\phi, \theta) = (\phi_f, \phi_h, \phi_j, \theta_f, \theta_h, \theta_s)$ encapsulate learnable DNN parameters of each function. The system efficiency is measured by the channel bandwidth ratio (CBR) $\rho = k/m$. The key of NTSCC lies in the variable-length deep JSCC guided by the latent prior on the semantic feature space. The latent prior $p_{y|z}(y|z)$ is obtained as

$$p_{y|z}(y|z) = \prod_i \frac{N(y_i|\mu_i, \sigma_i^2) \ast U(-\frac{1}{2}, \frac{1}{2})}{p_{z|y}(z|y_i)}$$

where the convolutional operation “$*$” with a standard uniform distribution is used to match the prior to the marginal such that the estimated rate $-\log p_{y|z}(y|z)$ is non-negative. Different from the data compression where the uniformly-noised proxy $y_i$ is applied in (1), we do not need the quantization operation such that the raw $y_i$ can be used. The hyperprior $z$ is usually transmitted over the digital link as side information due to its small cost, where the quantization $\lfloor \cdot \rfloor$ (rounding to integers) is needed as depicted in Fig. 1. A uniformly-noised proxy $\tilde{z} = z + \epsilon$ is used to replace the quantized representation $\tilde{y} = \lfloor y \rfloor$ during model training [23], where $\epsilon$ is sampled from $U(-\frac{1}{2}, \frac{1}{2})$. The probability of hyperprior $\tilde{z}$ is calculated on the fully factored density $p_z = \prod_j p_{z|y}(z|y_j)$ as

$$p_{z}(\tilde{z}) = \prod_j \left( \frac{p_{z|\psi(\cdot)}(\tilde{z}_j|\psi(\cdot)) \ast U(-\frac{1}{2}, \frac{1}{2})}{p_{\psi(\cdot)}} \right)$$

where $\psi(\cdot)$ encapsulates all the parameters of $p_{z|\psi(\cdot)}$.

The optimizing problem of NTSCC is formulated following the variational inference context [29], the posterior distribution $p_{\hat{s}, \hat{z}|x}$ is approximated using the variational density $q_{\hat{s}, \hat{z}|x}$ by minimizing their Kullback-Leibler (KL) divergence over the data distribution $p_x$ and the CSI distribution $p_h$ as the equation (11) in [23]. Accordingly, the optimization of NTSCC system can be formally converted to the minimization of the expected channel bandwidth cost, as well as the expected distortion of the reconstructed data versus the original, which leads to the optimization of the following RD tradeoff,

$$L_{RD}(\phi, \theta, \psi) = \mathbb{E}_{x \sim p_x} \mathbb{E}_{h \sim p_h} D_{KL}(q_{\hat{s}, \hat{z}|x} || p_{\hat{s}, \hat{z}|x}) \leftrightarrow \mathbb{E}_{x \sim p_x} \mathbb{E}_{h \sim p_h} \left( \lambda \left( -\eta_y \log p_{y|z}(y|z) - \eta_z \log p_z(\tilde{z}) \right) + d(\hat{x}, x) \right)$$

where the Lagrange multiplier $\lambda$ on the total channel bandwidth cost determines the tradeoff between the wireless transmission rate $R$ and the end-to-end distortion $D$. The scaling factors $\eta_y$ and $\eta_z$ are from the estimated entropy to the channel bandwidth cost and tied with the source-channel code capability and the wireless channel state. A larger $\eta_y$ indicates a better performance on deep JSCC codec $f_c$ and for $f_d$, but incurs more channel bandwidth cost. Accordingly, $\eta_y$ can be adjusted as a hyperparameter to control the system RD tradeoff. $\eta_z$ is not adjusted manually since explicit entropy coding and LDPC coding are selected to transmit the side information.

In practice, each embedding $y_i$ is a $c$-dimensional feature vector. The learned entropy model $-\log p_{y|z}(y_i|z)$ indicates the summation of entropy along $c$ dimensions of $y_i$, thus, the
information density distribution of $y$ is captured. Accordingly, the bandwidth cost, such as the number of OFDM subcarriers, $k_i$ for transmitting $y_i$ can be determined as

$$k_i = Q(k_i) = Q\left(-\eta_y \log p_{y_i|x}(y_i|z)\right),$$

(5)

where the learned entropy model $p_{y_i|x}$ follows $^{[4]}$, $Q$ denotes a scalar quantization whose range includes $2^q$ ($q = 1, 2, \ldots$) integers, and the quantization value set $\mathcal{V} = \{v_1, v_2, \ldots, v_{2^q}\}$ is related to the scaling factor $\eta_y$ and the Lagrange multiplier $q$. Hence, the predetermined $q$ bits should be transmitted as extra side information to inform the receiver which bandwidth is allocated to every embedding $y_i$. To adaptively map $y_i$ to a $k_i$-dimensional channel-input vector $s_i$, the dynamic neural network structure $^{[47]}$ is introduced into Transformers $^{[48]}$ to realize the deep JSCC codec $f_c$ and $f_d$ $^{[23]}$. By this approach, our NTSCC system is versatile: a single model can support multiple transmission rates by simply adjusting the scaling factor $\eta_y$, its rate tokens, and dimension-controlling FC (fully-connected) heads in deep JSCC $f_c$ and $f_d$ modules $^{[23]}$. The other parts, including nonlinear transforms $g_a, g_s, h_a, h_s$, and Transformer backbones in deep JSCC codec, stay unchanged. This versatile property brings clear advantage, one does not need to use different $\lambda$ values to train various models for achieving different RD tradeoffs. In short, we have a versatile semantic communication scheme.

### B. Motivation of Adaptive Semantic Communications

In existing works, the loss function $\mathcal{L}_{RD}$ in $^{[4]}$ is optimized over a corpus of source data samples (such as a large amount of images) and channel states in order to find optimal codec function parameters $\Sigma = (\phi, \theta, \psi)$. Although the models have been trained over a large corpus of source and channel samples for finding out what should be ideally optimal codec functions ($g_a, g_s, h_a, h_s, f_c, f_d$) over the whole data set and ergodic channel responses, we will show the codec functions can still be improved for each single data sample and instant CSI. This suboptimality of NTSCC model can be interpreted from the inference suboptimality of VAE $^{[30]}$ as follows: the mismatch between the true and approximate posterior. It has been proven that the inference gap includes two components: the approximation gap and the amortization gap. This approximation gap comes from the inability of the variational distribution family to exactly match the true posterior, and the amortization gap refers to the difference caused by amortizing the variational parameters over the entire training set, instead of optimizing for each training example individually.

Specifically, in our NTSCC system, given the source data sample $x$ and the instant CSI vector $h$, the inference gap $G$ is

$$G = D_{KL}(q_{\tilde{s}, z|x} \parallel p_{\tilde{s}, z|x}),$$

(6)

where $q_{\tilde{s}, z|x}$ is derived using the parameters $(\phi^*, \theta^*, \psi^*)$ learned under the entire training set, e.g.,

$$q_{\tilde{s}, z|x} = \arg\min_{q \in \mathcal{Q}} \mathbb{E}_{x \sim p_x} \mathbb{E}_{h \sim p_h} D_{KL}(q_{\tilde{s}, z|x} \parallel p_{\tilde{s}, z|x}).$$

(7)

However, the optimal $q_{\tilde{s}, z|x}^*$ should be derived under the given $x$ and $h$ as

$$q_{\tilde{s}, z|x}^* = \arg\min_{q \in \mathcal{Q}} D_{KL}(q_{\tilde{s}, z|x} \parallel p_{\tilde{s}, z|x}),$$

(8)

which corresponds to the optimal parameters $(\phi^*, \theta^*, \psi^*)$. As illustrated in Fig. 3, it can be derived by

$$G = D_{KL}(q_{\tilde{s}, z|x} \parallel p_{\tilde{s}, z|x}) = D_{KL}(q_{\tilde{s}, z|x}^* \parallel p_{\tilde{s}, z|x}) +$$

$$D_{KL}(q_{\tilde{s}, z|x}^* \parallel p_{\tilde{s}, z|x}) - D_{KL}(q_{\tilde{s}, z|x} \parallel p_{\tilde{s}, z|x}) = G_{\text{app}} + G_{\text{amo}}.$$

(9)

This indicates that the amortized posterior $q$ over the entire source dataset and ergodic CSI still incurs the amortization gap $G_{\text{amo}}$ under the instant $x$ and $h$. The above analysis demonstrates that a neural-network-based semantic communication model is trained on the entire dataset and CSI set with the target of achieving the best transmission RD performance on test data and CSI, i.e., we ideally expect $G_{\text{amo}} = 0 \iff G = G_{\text{app}}$. However, due to limited model capacity, optimization difficulties, and insufficient data and CSI, the model cannot in general achieve this goal. When the source data distribution or channel model differs from that in the training phase, model generalization will not be guaranteed even with the infinite data and model capacity, and perfect optimization.

We however note that a convenient feature of neural wireless data transmission is a model or semantic latent representation that can be easily finetuned on new data and CSI. A model can for instance or domain be trained after deployment. Inspired by this, as illustrated in Fig. 3, our goal is closing the amortization gap $G_{\text{amo}}$ by overfitting the source and channel such that the resulting posterior $q_{\tilde{s}, z|x}^*$ of our new adaptive NTSCC model can approach the optimal $q_{\tilde{s}, z|x}^*$ at every test instance. Thus the new amortization gap $G_{\text{amo}}^*$ is much smaller than $G_{\text{amo}}$. As such, we are effectively trying to solve the following optimization problem, given the instant $x$ and $h$.

$$(\phi^*, \theta^*, \psi^*) = \arg\min_{\phi, \theta, \psi} D_{KL}(q_{\tilde{s}, z|x} \parallel p_{\tilde{s}, z|x})$$

$$= \arg\min_{\phi, \theta, \psi} \mathcal{L}_{RD}(\phi, \theta, \psi, x, h) = \arg\min_{\phi, \theta, \psi} \left(\lambda \left(-\eta_y \log p_{y|x}(y|z) - \eta_z \log p_{z|x}(z)\right) + d(x, \hat{x})\right),$$

(10)

where $\mathcal{L}_{RD}(\phi, \theta, \psi, x, h)$ is the transmission RD objective for...
a particular $x$ and $h$.

III. OVERFITTING THE SOURCE

In this section, we present two methods to overfit a single source data instance $x$ (e.g., an image) or data from a specific domain $X'$ (e.g., a set of I-frames from a single video in the same scene). They stem from two different ideas:

- **Transmitter adaptation**: Given each $x$ or samples from $X'$, online update $(g_a, f_c)$ or $(y, s)$ using gradient descent applied on a baseline model learned on the entire dataset. This scheme is referred as “Tx-adapt”.

- **Transceiver adaptation**: Given each $x$ or samples from $X'$, online update $(g_a, f_c, f_s, g_s)$ using gradient descent applied on a baseline model learned on the entire dataset. This scheme is referred as “TxRx-adapt”.

Fig. 4 presents the key idea of overfitting the source in the NTSCC based ASC system.

Given the source sample $x$ and the baseline model $(\phi, \theta, \psi)$ learned on the entire dataset, our goal is adapting the transmitter model parameters $(\phi_g, \phi_f)$ or the latent code and channel-input codeword $(y, s)$ to every single data sample. The key benefit of transmitter adaptation is achieving an improved end-to-end transmission performance while keeping the predictive model fixed such that the computing time at the receiver stays unchanged. The refined nonlinear analysis transform $g_a$ or the latent code $y$ provides a more compact semantic representation of the source sample $x$, resulting in an improved transmission RD performance.

For model adaptation, we aim at effectively solving the following optimization problem. During the model inference time, for a single data instance $x$:

$$
(\phi_g^*, \phi_f^*) = \arg \min_{\phi_g, \phi_f} L_{RD}(\phi, \theta, \psi, x, h),
$$

where we adapt $g_a$ and $f_c$ while fixing the entropy model $h_a$ and $h_s$. This design aims to reduce the model updating complexity. In addition, if the side information $\bar{z}$ is transmitted, our method ensures that no model updates have to be transmitted to update $h_s$ at the receiver. Experiments can verify that our simplified method achieves comparable performance as that of finetuning the whole transmitter model $\phi = (\phi_g, \phi_f, \phi_h)$. In this work, we solve this problem (12) in an iterative procedure as that in [32]. We apply gradient descent on $\phi_g$ and $\phi_f$ to update the nonlinear analysis transform $g_a$ and the deep JSCC encoder $f_c$. The iterative parameter updating procedures are

$$
\phi_g^{(t)} = \phi_g^{(t-1)} - \gamma \nabla_{\phi_g} L_{RD}(\phi, \theta, \psi, x, h),
$$

$$
\phi_f^{(t)} = \phi_f^{(t-1)} - \gamma \nabla_{\phi_f} L_{RD}(\phi, \theta, \psi, x, h),
$$

where $\gamma$ denotes the learning rate. The final adaptive NTSCC pipeline is described in Algorithm 1.

The model adaptation can not only be applied for instance adaptation, but also be employed for domain adaptation. In
Algorithm 1: NTSCC with Tx Model Adaptation
Input: Baseline model parameters \((\phi, \theta, \psi)\) trained on training set, the data sample to be transmitted \(x\), the CSI vector \(h\).
Output: The reconstructed data sample \(\hat{x}\).

1: Initialize model parameters: \(\phi^0 \leftarrow \phi^f\).
2: for \(t = 1, 2, \ldots, T_{\text{max}}\) do
3: Initialize latent representation: \(s^0 \leftarrow \phi^f\).
4: Forward pass: \(x \xrightarrow{g_a(\cdot)} y \xrightarrow{f_c(\cdot)} s \xrightarrow{W(\cdot)} \hat{x}\) with the latent prior \(y \xrightarrow{h_a(\cdot)} z \xrightarrow{h_s(\cdot)} \{\mu, \sigma}\) and compute loss \(L_{\text{RD}}\) according to (10); update \(\phi^f\) using gradients \(\nabla_{\phi^f} L_{\text{RD}}\) as (13).
5: return Updated models \(\phi^f_1 \leftarrow \phi^f(T_{\text{max}})\) and \(\phi^f_{\text{max}}\).

8: procedure MODEL INERENCE\((\phi, \theta, \psi; x, h)\)
9: \((\phi^f_1, \phi^f_{\text{max}}) = \text{UPDATEMODELS}(\phi, \theta, \psi; x, h)\);
10: Encode: \(x \xrightarrow{g_a(\cdot)} y \xrightarrow{f_c(\cdot)} s\) with the latent prior \(y \xrightarrow{h_a(\cdot)} z \xrightarrow{h_s(\cdot)} \{\mu, \sigma}\);
11: Transmit over wireless channel: \(s \xrightarrow{W(\cdot)} \hat{s}\);
12: Decode: \(\hat{x} \xrightarrow{f_c(\cdot)} \hat{y} \xrightarrow{g_a(\cdot)} \hat{x}\); return Reconstructed data sample \(\hat{x}\).

Algorithm 2: NTSCC with Tx Code Adaptation
Input: Baseline model parameters \((\phi, \theta, \psi)\) trained on training set, the data sample to be transmitted \(x\), the CSI vector \(h\).
Output: The reconstructed data sample \(\hat{x}\).

1: Initialize latent representation: \(y^0 \leftarrow y\).
2: for \(t = 1, 2, \ldots, Y_{\text{max}}\) do
3: Forward pass: \(y^{(t-1)} \xrightarrow{f_c(\cdot)} s \xrightarrow{W(\cdot)} \hat{s} \xrightarrow{g_a(\cdot)} \hat{x}\) with the latent prior \(y^{(t-1)} \xrightarrow{h_a(\cdot)} z \xrightarrow{h_s(\cdot)} \{\mu, \sigma}\); compute loss \(L_{\text{RD}}\) according to (10); update \(y^{(t)}\) using gradients \(\nabla_{\theta} L_{\text{RD}}\) as (14).
4: return Updated latent representation \(y^* \leftarrow y^{(Y_{\text{max}})}\).

8: procedure UPDATE CODEWORDS\((\phi, \theta, \psi; x, h, s)\)
9: Initialize deep JSCC codeword: \(s^0 \leftarrow s\);
10: for \(t = 1, 2, \ldots, S_{\text{max}}\) do
11: Forward pass: \(s^{(t-1)} \xrightarrow{W(\cdot)} \hat{s} \xrightarrow{f_c(\cdot)} \hat{y} \xrightarrow{g_a(\cdot)} \hat{x}\); compute loss \(L_{\text{RD}}\) according to (10); update \(s^{(t)}\) using gradients \(\nabla_{\theta} L_{\text{RD}}\) as (15).
12: return Updated deep JSCC codeword \(s^* \leftarrow s^{(S_{\text{max}})}\).

15: procedure MODEL INERENCE\((\phi, \theta, \psi; x, h)\)
16: Encode: \(x \xrightarrow{g_a(\cdot)} y \xrightarrow{f_c(\cdot)} s\) with the latent prior \(y \xrightarrow{h_a(\cdot)} z \xrightarrow{h_s(\cdot)} \{\mu, \sigma\}\);
17: \(y^* = \text{UPDATELATENTS}(\phi, \theta, \psi; x, h, y)\);
18: Encode: \(y^* \xrightarrow{f_c(\cdot)} s\) with the latent prior \(y^* \xrightarrow{h_a(\cdot)} z \xrightarrow{h_s(\cdot)} \{\mu, \sigma\}\);
19: \(s^* = \text{UPDATE CODEWORDS}(\phi, \theta, \psi; x, h, s)\);
20: Transmit over wireless channel: \(s^* \xrightarrow{W(\cdot)} \hat{s}\);
21: Decode: \(\hat{x} \xrightarrow{f_c(\cdot)} \hat{y} \xrightarrow{g_a(\cdot)} \hat{x}\); return Reconstructed data sample \(\hat{x}\).

After \(t\) reaches \(Y_{\text{max}}\), the procedure turns to update \(s\), i.e.,
\[
\begin{align*}
  s^{(t)} &= s^{(t-1)} - \gamma \nabla_s L_{\text{RD}}(\phi, \theta, \psi; x, h), \\
  &\quad \text{with } t = 1, 2, \ldots, S_{\text{max}}.
\end{align*}
\] 

The total number of updating steps is \(T_{\text{max}} = Y_{\text{max}} + S_{\text{max}}\).

The final adaptive NTSCC pipeline is described by Algorithm 2. This latent representation and codeword adaptation method reduces the number of updated parameters for lower complexity, but it can only be applied for instance adaptation, but not for domain adaptation.

B. Transceiver Adaptation

The above transmitter adaptation method is appealing since no additional information needs to be added to the wireless transmitted signal, and nothing changes on the receiver. However, performance gains are quite limited since the deep JSCC...
decoder $f_d$ and the nonlinear synthesis transform $g_s$ cannot be adapted. In this subsection, we present a method for transceiver full-model adaptation, which adapts the entire NTSCC model to a single data instance or data from a specific domain. Unlike previous methods, our adaptive NTSCC with the full-model adaptation involves both content stream and model stream as the wireless transmitted signal. The model stream is employed to inform the receiving end to update the model parameters of $f_d$ and $g_s$. This yields a tripartite tradeoff among the data rate ($R$), model rate ($M$), and distortion ($D$) terms, formulating the transmission RDM loss.

Transceiver full-model adaptation online updates a set of global baseline model parameters $(\phi, \theta, \psi)$ on a single source data instance $x$. In practice, similar to that in the transmitter adaptation, we also fix the entropy model $h_o$ and $h_s$ to reduce the adaptation complexity. This results in the updated parameters $(\phi_o^*, \phi_s^*, \theta_o^*, \theta_s^*)$, of which only $\theta_o^*$ and $\theta_s^*$ are transmitted over the wireless channel as the model stream. Inspired by the residual coding idea, we only transmit the changes with respect to the baseline model $\delta_g = \theta_g^* - \theta_g$ and $\delta_f = \theta_f^* - \theta_f$ in practice. To encode the model updates $\delta = (\delta_g, \delta_f)$, we need to build a model prior $p_\delta(\delta)$ to quantify the model rate. Accordingly, the model rate is derived with $-\log p_\delta(\delta)$. Adding this term to the RD loss function in (18), we obtain the full-model instance-adaptive semantic transmission objective:

$$L_{RDM}(\phi, \theta, \psi, \delta, x, h) = L_{RD}(\phi, \theta + \delta, \psi, x, h) + \beta(-\eta_\delta \log p_\delta(\delta))$$

$$= \lambda \left( -\eta_\delta \log p_{y|x}(y|x) - \eta_\delta \log p_{z|x}(z) \right) + d(x, \hat{x})$$

$$+ \beta \left( -\eta_\delta \log p_\delta(\delta) \right),$$

where the scaling factor $\eta_\delta$ is tied with the capability of codec used to transmit $\delta$. $\delta$ denotes the reconstructed $\delta$ at the receiver end, and $\beta$ controls the tradeoff between the standard RD loss and the model transmission rate. Minimization of $L_{RDM}$ in (16) ensures any cost in the model stream contributes to the RD performance improvement.

For the model prior $p_\delta$, any probability distribution function can be selected, here, we naturally define $p_\delta$ as a factorized model, i.e., $p_\delta(\delta) = \prod_i p_{\delta_i}(\delta_i)$, where every component is of the same parameter. Each $p_{\delta_i}(\delta_i)$ is consistently generated from zero-centered Gaussian with a shared variance $\sigma^2$ as

$$p_{\delta_i}(\delta_i) = \prod_i \mathcal{N}(\delta_i | 0, \sigma^2) \ast (\Delta \cdot \mathcal{U}(\frac{-\Delta}{2}, \frac{\Delta}{2})) \mathcal{U}(\delta_i)$$

$$= \prod_i \int_{\delta_i - \frac{\Delta}{2}}^{\delta_i + \frac{\Delta}{2}} \mathcal{N}(\delta_i | 0, \sigma^2) \mathcal{U}(\delta_i) \, d\delta_i,$$

(17)

where the uniform distribution convolution is utilized to relax the prior such that the estimated model rate $-\log p_\delta(\delta)$ stays non-negative. It can directly interpolate the discrete probability values $p_{\delta_i}(\delta_i)$ at the quantized values $\delta_i$ that will be used when $\delta$ is transmitted over the digital link with entropy coding and channel coding, and $\Delta$ in (17) indicates the quantization bin width. If $\delta$ needs to be quantized for entropy coding, we define the quantization function as that in $\mathcal{N}$ with $N$ width-$\Delta$ quantization bins, i.e.,

$$\delta_i = \left\lfloor \frac{\delta - \Delta}{\Delta} \right\rfloor \cdot \Delta,$$

$$\min = -\frac{(N-1)\Delta}{2}, \max = \frac{(N-1)\Delta}{2}.$$
In can be observed that the value of instant CSI \(h\) is a key factor affecting the performance of deep JSCC codec. Under different channel states, different resource allocation strategies should be adopted to implicitly adjust source coding rate and channel coding rate inside the deep JSCC codec. Therefore, \(h\) is a deterministic factor for optimizing the deep JSCC modules \(f_c\) and \(f_d\). Existing learning approaches train the objective \([10]\) using different signal-to-noise ratio (SNR) indicating different channel states. Recently, an attention module was introduced into convolutional neural networks (CNN) to make deep JSCC SNR-adaptive with a single model \([19]\). But these methods cannot handle complex and varying wireless channels flexibly due to the following reasons:

- A simple average SNR cannot represent all properties of the wireless channel state \(h\).
- In practice, we usually assume the transmitter only has the feedback channel quality indicator (CQI) denoting the average SNR along all received symbols, but the receiver can get the CSI vector \(h\) by channel estimation indicating the SNR of each received symbol. Therefore, the simple use of average SNR to guide the design of both \(f_c\) and \(f_d\) is not realistic.

\[f \text{ deterministic factor for optimizing the deep JSCC modules } f_c, f_d \]

\[\Sigma(x, h') \Rightarrow \Sigma(x, h), \quad (23a)\]

\[\Sigma(x', h') \Rightarrow \Sigma(x', h). \quad (23b)\]
Our idea is introducing a model to every channel state instance \( h \) inside the deep JSCC codec [23]. For the proposed "Channel ModNet" is inserted after or before output of deep JSCC encoder in NTSCC as shown in Fig. 7. As illustrated in Fig. 2, the semantic feature map \( y \) output from the nonlinear analysis transform \( g_\phi(\cdot; \theta_e) \) is fed into the deep JSCC encoder \( f_e \). \( y \) consists of a group of \( l \) semantic feature vectors (embeddings) \( \{y_i|i = 1, 2, \ldots, l\} \). Each \( N \)-dimensional \( y_i \) is used as a token attached by a corresponding rate token into Transformer blocks. The output tokens \( \{y_i|i = 1, 2, \ldots, l\} \) are further fed into our ModNet to be modulated by the channel state information. Correspondingly, at the deep JSCC decoder, \( f_e \) is unreasonable for exploiting neural codec potentials.

- Online training the model according to every instant CSI \( h \) will lead to clearly increasing latency even though it brings the optimal performance. Thus, a flexible channel-dependent transmission method is required.

In this paper, to flexibly adapt our semantic communication model to every channel state instance \( h \), we develop a channel-dependent mechanism to enable the whole system automatically to adapt to \( h \) without relying on gradient descent. Our idea is introducing a plug-in module to modulate the output of deep JSCC encoder in NTSCC as shown in Fig. 7. The proposed "Channel ModNet" is inserted after or before Transformer blocks inside the deep JSCC codec [23]. For different channel states \( h \), we obtain different neural-syntax to generate a more specific deep JSCC codec functions \( f_e \) and \( f_d \).

In particular, the architecture of modified deep JSCC codec to overfit the instant channel state \( h \) is depicted in Fig. 7. As illustrated in Fig. 2, the semantic feature map \( y \) output from the nonlinear analysis transform \( g_\phi(\cdot; \theta_e) \) is fed into the deep JSCC encoder \( f_e \). \( y \) consists of a group of \( l \) semantic feature vectors (embeddings) \( \{y_i|i = 1, 2, \ldots, l\} \). Each \( N \)-dimensional \( y_i \) is used as a token attached by a corresponding rate token into Transformer blocks. The output tokens \( \{y_i|i = 1, 2, \ldots, l\} \) are further fed into our ModNet to be modulated by the channel state information. Correspondingly, at the deep JSCC decoder,
Channel ModNet includes 8 FC layers separated by 7 SNR modulation (SM) modules. SM is a three-layered FC network with input being the channel SNR that corresponds to $y_i$ (denoted as $\text{SNR}_i$). As stated before, the receiver can obtain the instant CSI $\text{h}$ via channel estimation such that the ModNet in the deep JSCC decoder $f_d$ can obtain the explicit $\text{SNR}_i$ for each $y_i$.

Herein, $\text{SNR}_i$ is computed by averaging the channel symbol SNRs along the $k_i$-dimensional sequence $s_i$. To be aligned with practical systems, we assume the CQI available at the transmitter via a feedback link. Herein, the CQI is an averaged SNR along all transmitted symbols $s$. Thus, the SM module input is $\text{SNR}_i = \overline{\text{SNR}}$ at the transmitter. The SM module transforms the input $\text{SNR}_i$ into an $N$-dimensional tensor $\text{sm}_i$ as depicted in Fig. 7. As such, the arbitrary target modulator can be realized by assigning a corresponding SNR value. The mapping procedures from $\text{SNR}_i$ to $\text{sm}_i$ are

$$\text{sm}_i^{(1)} = \text{ReLU}(W^{(1)} \cdot \text{SNR}_i + b^{(1)})$$

$$\text{sm}_i^{(2)} = \text{ReLU}(W^{(2)} \cdot \text{sm}_i^{(1)} + b^{(2)})$$

$$\text{sm}_i = \text{Sigmoid}(W^{(3)} \cdot \text{sm}_i^{(2)} + b^{(3)})$$

where Sigmoid is the activation function, ReLU denotes the rectified linear unit activation function, $\mathbf{W}^{(k)}$ and $\mathbf{b}^{(k)}$ are the affine function parameters and their bias. The CSI $\text{h}$ is therefore associated with the $N$-dimensional tensor $\text{sm}_i$ in each SM module. Then, the input $N$-dimensional feature will be fused with $\text{sm}_i$ in the element-wise product, i.e.,

$$\text{output} = \text{input} \odot \text{sm}_i.$$  

Here, input denotes the feature output from the previous FC layer, and output is feeding into the next FC layer.

Multiple SM modules are cascaded sequentially in a coarse-to-fine manner in Fig. 7. The previous modulated features are fed into subsequent SM modules. In our design, different ModNets are applied to different tokens $\{\hat{y}_i^t\}$, each token $\hat{y}_i^t$ is of the same dimension $N$. These tokens $\{\hat{y}_i^t\}$ are further sent into our Channel ModNet. The resulting modulated tokens will be fed into Transformer blocks to recover the semantic feature map $\hat{y}$. Thus, our ModNet is inserted into both encoder and decoder to modulate the intermediate tokens according to the instant wireless channel state.

V. EXPERIMENTAL RESULTS

In this section, we provide experimental results (including user study) to quantify the gains of our ASC system versus SOTA engineered communication systems. Then, critical discussions about results and our adaptation method are given.

A. Experimental Setup

1) Datasets: Regarding the wireless video transmission, we adopt the Video Streaming Dataset 4K (VSD4K) dataset [36] to verify the performance of our ASC paradigm, and VSD4K dataset is composed of several 4K 30fps videos, including six popular video categories: game, vlog, interview, sport, dance, and city. The first three categories are mainly single-scene but have multiple points of view, while the latter contains various scenes and large-scale motion. In this paper, we consider the I-frame wireless transmission problem, i.e., the test set consists a group of I-frame images from a single video of a specific category, which is a subproblem of video transmission. In particular, we select a 45s-length representative video from each category, subsample each video by the first of four frames to create a dataset of I-frames, and generate low-resolution (LR) videos of 480P, 720P, and 1080P using bicubic interpolation. Unless otherwise specified, the following experiments will be carried out on 480P video sources.

2) Network Implementation: Note that the proposed source and channel overfitting method are general that can be transplanted to any neural network based semantic communication system, we realize adaptive NTSCC as a representative of the ASC system in this paper. As stated before, we do not transmit the side information $\tilde{x}$ to simplify the NTSCC based semantic communication system implementation [23]. We use the DIV2K [51] image set to train our baseline model, which contains 800 natural images of 2K resolutions on average. The baseline model training procedure includes 6,000 epochs using Adam optimizer [52] with the learning rate $\gamma = 1 \times 10^{-4}$. During model training, the images are randomly cropped to $256 \times 256$ patches to form a batch of 8 patches. In all experiments, we train four baseline models with $\lambda = 256, 64, 16, 4$ respectively under the additive white Gaussian noise (AWGN) channel, where the scaling factor $\eta_y$ is 0.2 at SNR = 10dB and 0.4 at SNR = 0dB. The mean squared error (MSE) is used as the distortion measurement to qualify the end-to-end image transmission performance. Besides, for subjective comparison, we further consider perceptual quality optimization to better align with the human vision in semantic communication system. The training details of NTSCC (Perceptual) are the same as that in [23].

3) Transmitter Adaptation Details: Our Tx-adapt schemes include the transmitter model adaptation and the transmitter code adaptation. The former can be applied to either every I-frame (instance adaptation) or every video sequence (domain adaptation) in VSD4K dataset, but the latter can work only in the instance adaptive mode. Specifically, in the case of the instance adaptation, the transmitter model adaptation scheme updates $(g_a, f_c)$ for $T_{\text{max}} = 100$ steps per instance with the learning rate $\gamma = 1 \times 10^{-4}$. In the case of transmitter code adaptation, we first update $y$ for $Y_{\text{max}} = 50$ steps and then update $s$ for another $S_{\text{max}} = 50$ steps, where the learning rate is set to $\gamma = 1 \times 10^{-5}$ in both optimization procedures. One
4) Transceiver Adaptation Details: Our adaptive NTSCC with TxRx-adapt starts from the aforementioned four baseline models, we online update the parameters \((\phi_g, \phi_f, \theta_g, \theta_f)\) for the RDM loss \(L_{\text{RDM}}\) over each video sequence. To make the average channel bandwidth cost of updated models close to the baseline, we set the RDM tradeoff hyperparameter tuples \((\lambda, \beta)\) as \((256, 4), (32, 4), (4, 1),\) and \((2, 1)\), which correspond to the baseline models learned with \(\lambda = 256, 64, 16, 4\), respectively. For model updates (model stream), we set the parameter quantization bin width as \(\Delta = 0.005\) and adopt the GMM prior, where \(\sigma\) in \(g_{\text{lab}}\) is set to 0.05, and the spike weight \(\alpha = 1000\). The number of updating steps is set to \(T_{\text{max}} = 10000\), with the learning rate of \(\gamma = 1 \times 10^{-5}\) for \(g_a\) and \(f_c\), and \(\gamma = 1 \times 10^{-4}\) for \(g_s\) and \(f_d\). We select the best updated model based on the RDM loss for model inference.

5) Comparison Schemes: Apart from the comparison with baseline NTSCC system, we also compare our adaptive NTSCC system with SOTA engineered wireless video transmission systems. In particular, we adopt HEVC (H.265) [28] and VVC (H.266) [29] for video compression combined with 5G LDPC code [22] for channel coding. We use \("+\)" to combine source coding and channel coding schemes for brevity. For VVC and HEVC, we use the official test model VTM-12.2 with intra profile and BPG software to encode I-frames in the YUV444 mode, and calculate the PSNR in RGB. Besides, the emerging NTSC source compression scheme in [53] combined with 5G LDPC codes are also included in comparison. We adopt the released NTC models based on mean and scale hyper-prior implemented by CompressAI [54]. All I-frames are cropped to multiples of 64 to avoid padding for neural codecs for a fair comparison. The above simulations are implemented on the top of Sionna [55], which is an open-source library for link-level simulations of digital communication systems. Apart from these, we also compare our adaptive NTSCC with the emerging neural deep JSCC transmission scheme [14]. For fair comparison, the channel bandwidth cost counts up all the transmitting streams over the wireless channel, including both data and model in our full-model adaptive NTSCC.

B. Results of Overfitting the Source

We first consider the domain adaptation case. Fig. 8(a)–(h) show the averaged RD performance under AWGN channels with \(\text{SNR} = 0\) dB (bad channel condition) and \(\text{SNR} = 10\) dB (good channel condition), respectively. In Fig. 8 the legend “NTSCC (TxRx-adapt)” denotes NTSCC with our transceiver full-model adaptation, and the “NTSCC (Tx-adapt)” denotes our transmitter model adaptation. For the classical separation-based schemes, according to the adaptive modulation coding (AMC) principle, we need to traverse given combinations of LDPC coded modulation schemes to find the highest-efficiency scheme under the reliable transmission constraint (block error rate \(\leq 10^{-5}\)). Accordingly, we adopt a 1/3 rate (2048, 6144) LDPC code with 4QAM at \(\text{SNR} = 0\) dB, and a 2/3 rate (4096, 6144) LDPC code with 16-ary quadrature amplitude modulation (16QAM) at \(\text{SNR} = 10\) dB. Results in Fig. 8 indicate that by using our proposed online overfitting mechanisms, we achieve considerable gains compared to the baseline NTSCC semantic communication system, and the transceiver full-model adaptation results in higher performance gain than updating the transmitter only. Moreover, compared to the emerging best performed “VTM + 5G LDPC” transmission...
system, our adaptive NTSCC semantic communication system (TxRx-adapt) shows comparable or even better performance. Therefore, as one typical scheme of ASC, our adaptive NTSCC shows the potential becoming the emerging SOTA scheme for end-to-end wireless communications. The existing well-known improvements of semantic communication are compared with “BPG + 5G LDPC” [23] or on the tiny-scale image dataset [13], [14], our performance gain is more meaningful as the first semantic communication system overpassing the SOTA coded transmission system (VTM + 5G LDPC) on high-resolution image/video datasets.

It is worth noting that, according to the JVET report [56], VVC (H.266) is 11.3 times more complex than HEVC (H.265) in exchange for a 25% performance gain, and the encoding time of VVC I-frame coding is 25% in exchange for a 41% bandwidth cost, while NTSCC with only transmitter adaptation (Tx-adapt) can save up to 13%. Among the six video sequences in VSD4K, our NTSCC (Tx-adapt) system performs better than all other systems for most categories (4/6) while indeed performing slightly worse on sport and vlog sequences. We observe that these two sequences include multiple shot and diverse scenarios, which lead to performance

We further plot the PSNR gain or bandwidth saving of six sequences in VSD4K dataset in Fig. 9. Here, we employ the widely-used Bjøntegaard Delta (BD) rate reduction algorithm [58] for relative performance evaluation. The baseline scheme is the NTSCC semantic communication system learned on the entire dataset [23]. Fig. 9(a) and 9(b) show the average CBR percentage over the baseline at the same PSNR under SNR = 0dB and SNR = 10dB, respectively. Fig. 9(c) and 9(d) show the average quality improvement in terms of PSNR over the baseline for each scheme. In particular, our proposed NTSCC with transceiver adaptation (TxRx-adapt) can save up to 141% bandwidth cost, while NTSCC with only transmitter adaptation (Tx-adapt) can save up to 13%. Among the six video sequences in VSD4K, our NTSCC (TxRx-adapt) system performs better than all other systems for most categories (4/6) while indeed performing slightly worse on sport and vlog sequences. We observe that these two sequences include multiple shot and diverse scenarios, which lead to performance
loss by adapting the baseline NTSCC model to the whole video sequence. We should restrict our adaptation range to match better with the obvious scene changes within one video sequence.

To explore the effect of domain adaptation range, we select three clips from the sport sequence and plot the cumulative distribution function (CDF) of PSNR gains versus the baseline NTSCC semantic communication system learned on the entire dataset. The results are provided in Fig. 10. We further restrict our adaptation range to each scene on the whole sport sequence (subsequence domain adaptation). In this case, our full-model adaptive NTSCC (TxRx-adapt) requires more model stream bandwidth cost (reflecting as the instantaneous decrease of PSNR gain at scene transition time in Fig. 10), but this extra cost to transmit decoder model updates will be soon amortized and covered with time, which leads to superior performance than other instance or domain adaptation schemes. That means the performance loss of our adaptive NTSCC (TxRx-adapt) versus VTM + 5G LDPC in Fig. 8 and Fig. 9 can be efficiently reduced by restricting the domain adaptation range. It shows as the superiority of “subsequence TxRx-adapt” versus “sequence TxRx-adapt” in Fig. 10. Besides, we observe that the two transmitter instance adaptive methods (Tx-adapt model, and Tx-adapt code) show different performance, but they both outperform the domain adaptive methods used in Fig. 11 and Fig. 9. Herein, the transmitter model adaptation (updating \( g_e \) and \( f_e \)) spends more computational complexity for better performance.

Next, we discuss the tripartite tradeoff among the data rate (\( R \)), model rate (\( M \)), and distortion (\( D \)). In Table I, we present the distribution of CBR cost using a fixed \( \lambda = 4 \) in the game sequence. With the decrease of \( \beta \), the relative contribution of model stream increases, which leads to the reduction in the amelioration gap \( g_{sam} \) as analyzed in Fig. 3. Consequently, the performance comes better in terms of the tradeoff between content rate and end-to-end distortion. Since the small model stream cost can be amortized over a great many of I-frames, the amount of model rate is indeed negligible especially in the high \( \beta \) region. As a result, compared to the baseline model (listed as the last line in Table 1), an appropriate selection of \( \beta \) in the transceiver adaptation can contribute to significant improvement on the reconstruction quality while also reducing the total rate \( R + M \).

Fig. 11 illustrates the sequence domain adaptation progression over time using different \( \lambda \) with a fixed \( \beta = 1 \). It clearly demonstrates that optimizing for different \( \lambda \) leads the RDM curve moving in different directions. We also find that the final RDM performance point is independent of the starting point from baseline model. This phenomenon indicates that given the model stream weight \( \beta \), the optimal amortization gap reduction on the receiver has already been determined. In order to further reduce the amortization gap, the only approach is to restrict the range of adaptation domain. Accordingly, the extreme instance adaptation can result in the smallest amortization gap theoretically.

We further report the performance on the game video at different resolutions over the AWGN channel at \( \text{SNR} = 10 \text{dB} \), where the average channel bandwidth cost is computed within 100ms according to the 5G OFDM configurations: 14 OFDM symbols@1ms with the subcarrier bandwidth 15KHz.

To intuitively demonstrate the effect of our source overfitting method, we further pick visible results on transmitting the city video sequence in Fig. 13. In this figure, we also show...
Fig. 13. Examples of visual comparison. The first column shows the original image and its cropped patch. The second to the fifth column show the reconstructed images by using different transmission schemes over the AWGN channel at SNR = 0dB, where the metrics in parentheses indicate the model training target loss function (PSNR or perceptual loss). Red number and blue number indicate the percentage of bandwidth cost increase and saving compared to the “VTM + 5G LDPC” scheme.

Fig. 14. User study results. As an anchor, the black diamond point denotes the CBR cost of our adaptive NTSCC (TxRx-adapt) semantic communication system optimized under perceptual loss. For each data point in the red line, its Y-axis shows what percentage of users prefer our scheme, and the X-axis is the CBR cost of VTM + 5G LDPC. The blue arrow highlights how much extra CBR cost VTM + 5G LDPC spends where still more than 50% of participants prefer ours. The results are counted from 2953 ratings in total, with an average of 123 per data point.

our adaptive NTSCC model updated under perceptual loss as that in [23] and [59]. From these results, we can observe that our adaptive NTSCC semantic transmission scheme optimized under the perceptual loss [59] shows much better visual quality with lower channel bandwidth cost. Thus, it can better support the human vision demands in semantic communications.

To better aligned with human perception, we have conducted the user study. We employ the “two alternatives, forced choice” (2AFC) test for quantitative evaluation. As a widely-used approach in perceptual visual quality assessment [5], [59], [60], the user study interface shows human raters a group of three images in the same column, where the middle image is always the original image, and the other two are lossily generated from two different methods. Then, we require raters to choose which image (left or right) looks closer to the original. In our semantic communication scenarios, we adopt the adaptive NTSCC (TxRx-adapt) model trained under perceptual loss [23] with $\lambda = 64$ and compare it to VTM + 5G LDPC with close or more CBR cost. In particular, we randomly select 10 frame images for each video sequence transmitted over AWGN channel at SNR = 0dB and SNR = 10dB, respectively. The entire rating process consists of several rounds in which all participants were asked to vote on each of the 10 groups of images. If our adaptive NTSCC (TxRx-adapt) dominates, the CBR cost of VTM + 5G LDPC will be increased in the next round until it receives more votes. The results of the user study are shown in Fig. 14. As we can see, our source and channel overfitting approach enables NTSCC to greatly outperform the SOTA VTM + 5G LDPC transmission scheme, especially in the game and city sequences. It verifies that the proposed
ASC system can better support the human perceptual vision demands in wireless data transmission, which is aligned with the target of semantic communications.

C. Results of Overfitting the Channel

This subsection verifies the flexibility of our proposed plug-in Channel ModNet to overfit the instant channel state. We consider both the fundamental AWGN channel and the widely-used COST2100 wireless channel model [61].

Fig. 15 shows the image transmission performance comparison as a function of channel SNR over the AWGN channel under the CBR constraint $\rho \leq 0.03$, where the results are tested on the Kodak dataset ($768 \times 512$ pixels) [62]. We use the dash lines to show the performance of baseline NTSCC models trained under SNR $= 10$dB and SNR $= 0$dB, respectively. Although our baseline NTSCC semantic communication schemes achieve graceful degradation instead of suffering cliff-effect like BPG + LDPC series, they still incur obvious performance loss when the testing channel SNR differs far from the training SNR. However, by plugging our designed Channel ModNet into the NTSCC (10dB model) and finetuning over SNR from $-5$dB to 20dB, we realize the channel-dependent transmission using one model to adapt to diverse channel conditions. Our channel-dependent NTSCC model achieves the performance of BPG combined with ideal channel capacity-achieving code. It verifies the efficiency of our channel overfitting method on the fundamental AWGN or the flat fading channels.

To further verify the effectiveness of our plug-in Channel ModNet, we carry out experiments on the practical COST2100 wireless fading channel [61]. CSI samples are collected in an indoor scenario at 5.3GHz bands. In this case, the transmitted symbols $s$ will pass over a frequency selective channel on the OFDM grid. We transmit the video sequence I-frames over this channel. Fig. 16 shows the results. By inserting the Channel ModNet module, our codec can be well aware of the instant embedding-wise SNRs. It is more accurate than using a rough average SNR among all embeddings. Therefore, our channel-dependent NTSCC performs better than the standard NTSCC model trained at a given average SNR. Furthermore, the performance can be further improved by full-model adaptation based on the channel-dependent model, as shown in the red line, which realizes source domain adaptation and channel instance adaptation. Besides, we also present a comparison with BPG + 5G LDPC schemes using the same CSI samples collected from COST2100. The configurations of LDPC codes and QAM are the same as that in Fig. 15 and we take the envelope of all combinations of coded transmission schemes as the final performance of BPG + 5G LDPC (black line in Fig. 16). Apparently, it cannot provide satisfactory quality due to the one-shot transmission under practical fading channel. To align better with practical communication system, we further adopt the hybrid automatic repeat request (HARQ) with chase combining (CC) [63] to enhance the system performance while our NTSCC based semantic communication still uses one-shot transmission. Retransmissions bring considerable PSNR gains, and the gain increases with the number of HARQ allowed, but HARQ introduces much higher latency. From the results, we observe that our NTSCC with one-shot transmission (red line in Fig. 16) still outperforms traditional schemes using HARQ, which is very meaningful for delay-sensitive services like XR or auto-driving.

D. Results Discussion

Regarding the experimental results, we can draw two conclusions:

- First, our online overfitting mechanism enables semantic communication system to efficiently adapt to both source content and wireless channel state. It can greatly reduce the model amortization gap, which finally contributes to the system performance gain.
- Second, the small additional complexity caused by model adaptation involves only at the transmitter. There is no extra complexity at the receiver, which is friendly to most content delivery tasks with the request of low latency. It is aligned with the evolution idea of both traditional and neural video codec.

In summary, as mentioned in Section 4, almost all traditional source compressors follow the hybrid transform coding
paradigm to evolve, e.g., mode selection in HEVC [28] and VVC [29]. Accordingly, the idea of signal-dependent transform in traditional source compression methods inspires us to upgrade the deep learning based semantic communication system to a content-channel-dependent adaptive mode. The marriage of deep learning method and traditional coding idea can bring significant performance improvement.

Based on previous works about semantic communications [13]–[20], [23], [24], and the work in this paper, we think that the advantages of neural network based end-to-end semantic communication system are three folds:

• First, the excellent content and semantics adaptivity of neural network is superior to signal processing based traditional models since the network parameters are learned based on lots of practical source and channel data samples while the models in the SOTA coded transmission standards are handcrafted based on prior knowledge.

• Second, the neural network can well represent and utilize source and channel features, which makes the semantic communication system can be optimized toward both human view perception and machine vision tasks. However, the existing source and channel coding standards only pursue high performance toward the objective quality assessment indices, e.g., PSNR.

• Third, the rate-distortion (RD) optimization guided neural network training and adaptive switching for semantic communication is quite effective and efficient. As analyzed in this paper, a single model to deal with all the source data with diverse structures and varying wireless channel states is inefficient obviously. Therefore, the adaptively learning and switching according to source data and channel state is a necessary solution to enhance the RD performance for all deep learning based semantic communication systems. In addition, compared with the adaptive coding paradigm used in traditional source and channel coding where one appropriate coding mode is selected from the predefined mode pool with limited number of options, our ASC system leverages the overfitting property of neural network to adapt to arbitrary coding mode. Hence, our RD guided learning ASC system is much more flexible to be tailored for specific source and channel instance, also, its adaptation complexity is lower than the mode selection in traditional coding methods.

In a nutshell, our overfitting method has efficiently catalyzed semantic communication system to provide more promising results. It is a necessary approach for all learning-based end-to-end communication system to further boost performance, which is indeed aligned with the evolution route of both traditional and neural video codec [64].

VI. CONCLUSION

In this paper, we have proposed a novel semantic communication paradigm by well leveraging the deep learning model’s overfitting property. Our model can for instance or domain be updated after deployment, which leads to substantial gains on the transmission rate-distortion (RD) performance. In this way, the emerging semantic communication is further upgraded to the adaptive semantic communication (ASC) system. Our ASC system is dual-functional for adapting to both source domain and CSI instance. Specifically, we have proposed several methods to update the semantic communication models. Unlike previous systems, the ingredients of wireless transmitted stream include both the semantic representations of source data and the adapted decoder model parameters. Accordingly, we have formulated the whole ASC design as an new optimization problem whose goal is minimizing the loss function that is a tripartite tradeoff among the data rate, model rate, and end-to-end distortion terms. Results have verified the substantial gains of our ASC system. As a new paradigm, our model adaptation method has the potential to catalyze semantic communication upgrading to a new era.

REFERENCES

[1] P. Zhang, W. Xu, H. Gao, K. Niu, and et al., “Toward wisdom-evolutionary and primitive-concise 6G: A new paradigm of semantic communication networks,” Engineering, vol. 8, pp. 60–73, 2022.
[2] H. Xie, Z. Qin, G. Y. Li, and B.-W. Jiang, “Deep learning enabled semantic communication systems,” IEEE Transactions on Signal Processing, vol. 69, pp. 2663–2675, 2021.
[3] J. Dai, P. Zhang, K. Niu, S. Wang, Z. Si, and X. Qin, “Communication beyond transmitting bits: Semantics-guided source and channel coding,” IEEE Wireless Communications, early access, 2022.
[4] C. E. Shannon, “A mathematical theory of communication, 1948,” Bell System Technical Journal, vol. 27, no. 3, pp. 3–55, 1948.
[5] R. Zhang, P. Isola, A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 586–595.
[6] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, “Image assessment: Unifying structure and texture similarity,” arXiv preprint arXiv:2004.07728, 2020.
[7] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” Advances in Neural Information Processing Systems, vol. 30, 2017.
[8] L. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, “Video coding for machines: A paradigm of collaborative compression and intelligent analytics,” IEEE Transactions on Image Processing, vol. 29, pp. 8680–8695, 2020.
[9] M. Fresia, F. Perez-Cruz, H. V. Poor, and S. Verdu, “Joint source and channel coding,” IEEE Signal Processing Magazine, vol. 27, no. 6, pp. 104–113, 2010.
[10] A. Guyader, E. Fabre, C. Guilletmot, and M. Robert, “Joint source-channel turbo decoding of entropy-coded sources,” IEEE Journal on Selected Areas in Communications, vol. 19, no. 9, pp. 1680–1696, 2001.
[11] N. Ramzan, S. Wan, and E. Izquierdo, “Joint source-channel coding for wavelet-based scalable video transmission using an adaptive turbo code,” EURASIP Journal on Image and Video Processing, vol. 2007, pp. 1–12, 2007.
[12] C. Chen, L. Wang, and F. CM Lau, “Joint optimization of protograph LDPC code pair for joint source and channel coding,” IEEE Transactions on Communications, vol. 66, no. 8, pp. 3255–3267, 2018.
[13] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, “Deep joint source-channel coding for wireless image transmission,” IEEE Transactions on Cognitive Communications and Networking, vol. 5, no. 3, pp. 567–579, 2019.
[14] D. B. Kurka and D. Gündüz, “DeepSCC-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.
[15] D. B. Kurka and D. Gündüz, “Bandwidth-agile image transmission with deep joint source-channel coding,” IEEE Transactions on Wireless Communications, vol. 20, no. 12, pp. 8081–8095, 2021.
[16] 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, 2020.
[17] N. Farsad, M. Rao, and A. Goldsmith, “Deep learning for joint source-channel coding of text,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2018, pp. 2326–2330.
