

Latent Space Alignment for AI-Native MIMO Semantic Communications

Mario Edoardo Pandolfo¹, Simone Fiorellino^{1,2}, Emilio Calvanese Strinati³, and Paolo Di Lorenzo^{2,4}

¹ DIAG Department, Sapienza University of Rome, via Ariosto 25, Rome, Italy.

² Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), Parma, Italy.

³ CEA Leti, University Grenoble Alpes, 38000, Grenoble, France.

⁴ DIET Department, Sapienza University of Rome, Via Eudossiana 18, Rome, Italy.

E-mail: {marioedoardo.pandolfo, simone.fiorellino, paolo.dilorenzo}@uniroma1.it,
emilio.calvanese-strinati@cea.fr.

Abstract—Semantic communications focus on prioritizing the understanding of the meaning behind transmitted data and ensuring the successful completion of tasks that motivate the exchange of information. However, when devices rely on different languages, logic, or internal representations, semantic mismatches may occur, potentially hindering mutual understanding. This paper introduces a novel approach to addressing latent space misalignment in semantic communications, exploiting multiple-input multiple-output (MIMO) communications. Specifically, our method learns a MIMO precoder/decoder pair that jointly performs latent space compression and semantic channel equalization, mitigating both semantic mismatches and physical channel impairments. We explore two solutions: (i) a linear model, optimized by solving a biconvex optimization problem via the alternating direction method of multipliers (ADMM); (ii) a neural network-based model, which learns semantic MIMO precoder/decoder under transmission power budget and complexity constraints. Numerical results demonstrate the effectiveness of the proposed approach in a goal-oriented semantic communication scenario, illustrating the main trade-offs between accuracy, communication burden, and complexity of the solutions.

Index Terms—Semantic Communication, semantic equalization, latent space alignment, MIMO, AI-native communication.

I. INTRODUCTION

Classic communication paradigms focus on the accurate transmission of individual symbols or bits. The rapidly increasing number of connected devices, combined with the sheer volume of information required by emerging applications, is rapidly overwhelming the capacity of traditional bit-centric communication systems [1]–[3]. For instance, in time-sensitive applications like autonomous driving, industrial automation, and smart surveillance, it is crucial to deliver reliable AI services with ultra-low latency while balancing energy efficiency, communication overhead, and computational capacity. Despite ongoing advancements, current wireless systems are increasing bandwidth and energy use to meet the growing demand for higher data rates. However, this pursuit faces resource constraints, including limited spectrum, energy, and

computational power, highlighting the need for a new communication paradigm for future applications [4]–[8].

A key paradigm shift for 6G systems and envisioned IoT applications lies in the adoption of semantic communications (SCs) [4], [6]. This approach moves beyond traditional bit-related metrics commonly used in current system design and optimization, instead prioritizing the interpretation of the meaning behind transmitted data and the successful completion of tasks that drive the information exchange [6], [7], [9], [10]. Taking this idea further, novel approaches to semantic joint source and channel coding [7], semantic extraction and compression [11], [12], goal-oriented system design and optimization [13]–[15], semantic reasoning [16], and semantic communications based on generative AI [17] have been proposed in the literature. A key enabler of SCs is artificial intelligence (AI), where deep neural networks (DNNs) are exploited to extract and transmit key semantic features from data rather than the raw data itself [7], [18]. At the receiver side, a DNN decoder reconstructs and interprets the intended message, offering inherent robustness to noise and imperfect transmission conditions. This approach was shown to reduce the amount of data that needs to be transmitted, thereby saving bandwidth and energy [7], [18]. In SCs, two primary sources of noise exist [8]: (i) *channel noise*, which alters the received symbols at the physical layer; (ii) *semantic noise*, which can arise from various causes, e.g., misalignment of logic, interpretation, or knowledge among communication devices. Both types of noise can significantly disrupt communication.

The handling of channel noise has historically been guided by Shannon’s Separation Theorem [19], which asserts that separate source and channel coding achieves optimality in the asymptotic regime of infinite block length and unbounded complexity. Although suitable for traditional communication networks, this approach has proven suboptimal in practical finite block-length scenarios, especially for emerging applications. Joint source-channel coding (JSCC) addresses these limitations by directly mapping source signals to channel symbols, bypassing intermediate bit representations. Recent advances in deep learning have led to practical deep learning-based JSCC (DeepJSCC) schemes [20]–[24], where DNNs are used to jointly optimize the source-channel coding problem in an end-to-end fashion. DeepJSCC shown to be suitable for

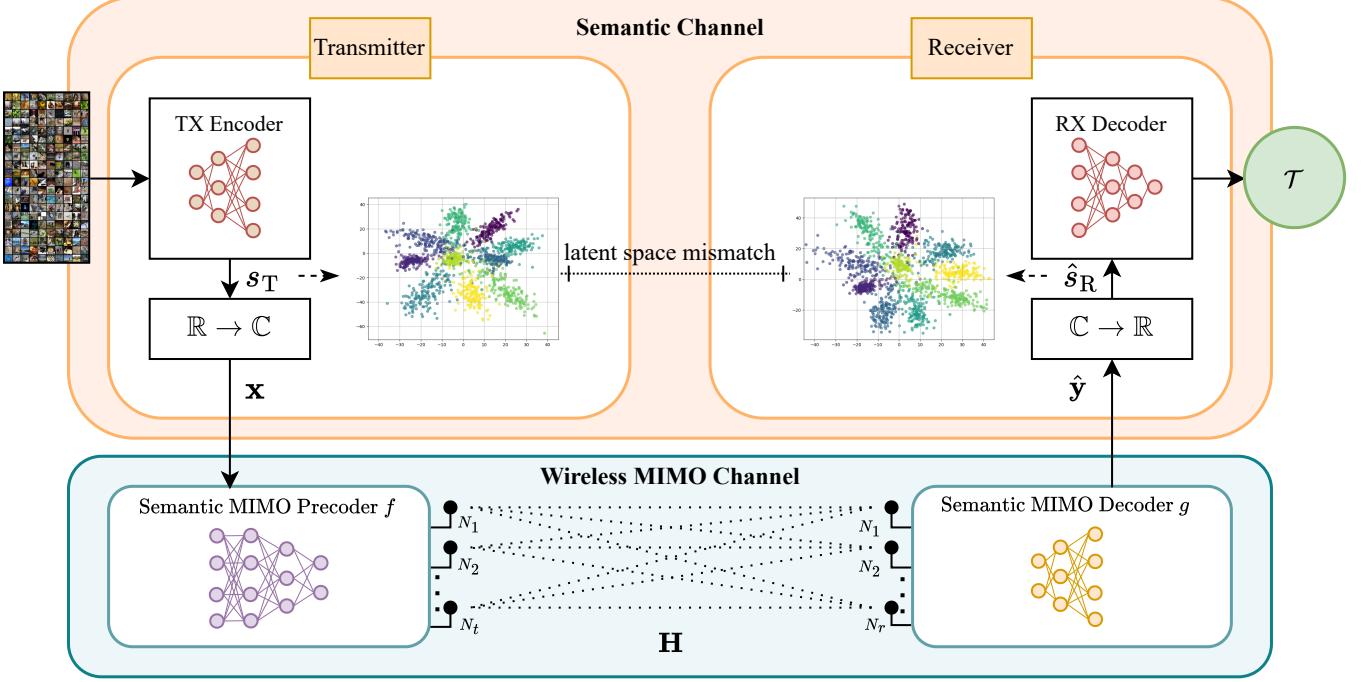


Fig. 1: System model: The semantic precoder f of TX encodes a compressed latent space s_i into a transmitter vector of dimension N_T . It is transmitted through the MIMO channel \mathbf{H} with noise \mathbf{v} . Then, the received vector is decoded by the semantic decoder g and decompressed to match the RX latent space.

enabling SCs schemes [7], [25] while performing both semantic compression and channel equalization (source-channel coding). However, most existing DeepJSCC frameworks assume consistent latent representations at transmitter and receiver, and therefore do not address the issue of semantic noise.

In this work, we specifically address the challenge of *latent space misalignment*, i.e., a type of semantic noise due to the heterogeneous ways used by DNNs at different devices to encode and interpret the exchanged semantic information. This problem, also known as *semantic channel equalization*, was investigated in some recent works [15], [26]–[32]. In [29], the authors introduce a semantic channel equalizer that models mismatches with measurable transformations over semantic spaces and applies tools from optimal transport theory. The work in [27] introduces relative representations (RRs) to perform (zero-shot) latent space communication among DNN models, and the paper in [15] exploits RRs in a goal-oriented communication context performing a dynamic optimization of edge resources. In [31], the authors leverage RRs to align models without the need to train a new decoder. The alignment process is achieved by computing the pseudo-inverse of the anchor matrix, and cosine similarity is used as the similarity function. In [32] is proposed a semantic equalization framework based on RRs that do not need any retraining and require sharing a small amount of data between transmitter and receiver. The method is agnostic to the similarity function of RRs. However, the works in [15], [27], [29], [31], [32] did not investigate the effect and, most importantly, the optimization of MIMO wireless communication for semantic channel equalization, thus leading to a full integration of AI-based semantic extraction and wireless aspects.

Contributions. In this work, we present an optimization framework for AI-native, MIMO, semantic, and wireless communications, addressing the impact of semantic noise modeled as latent space mismatch. Specifically, we model the MIMO communication as a chain of (learnable) transformations composed of a (semantic) precoder, the channel matrix, and a (semantic) decoder. Then, we perform a joint optimization of semantic precoding/decoding that minimizes a semantic performance metric of interest, promoting latent space alignment between semantic feature spaces at the transmitter and receiver sides. To trade-off complexity and performance, we explore two optimization solutions. The first assumes linear precoding/decoding modules, and the related problem is formulated as a biconvex program numerically solved through ADMM. In the second case, we assume nonlinear precoding/decoding modules modeled by DNNs, whose parameters are learned through a specific training procedure, which takes into account transmission power budget and complexity constraints. The proposed framework performs semantic compression and equalization jointly, while handling both semantic and channel noise effectively. Additionally, a detailed analysis is carried out to assess the computational complexity of the proposed DNN solution. Numerical results evaluate the effectiveness of our methodology in a goal-oriented semantic communication setting, where data exchange supports image classification, illustrating how our approach leads to better performance with respect to available benchmarks from the literature.

II. SYSTEM MODEL

We consider a communication system where the transmitter (TX) and the receiver (RX) employ (pre-trained) DNNs to

encode and decode semantic information, resulting in the transmission (and interpretation) of latent representations. A pictorial representation is illustrated in Fig. 1. Let $\mathbf{s}_T \in \mathbb{R}^d$ be the vector of semantic features extracted at the TX side from a data point $\mathbf{z} \in \mathbb{R}^q$. The set of all semantic vectors \mathbf{s}_T represents the TX semantic latent space. We assume that the RX is trained to interpret a different encoding scheme than the one used by the TX, i.e., it requires the reception of a different semantic feature vector $\mathbf{s}_R \in \mathbb{R}^m$ (corresponding to \mathbf{z}) to correctly interpret the transmitted message or effectively perform a given task (e.g., classification). The set of all semantic vectors \mathbf{s}_R represents the RX semantic latent space. Due to the mismatches between TX and RX latent spaces, the communication is prone to semantic noise and must be equalized to enable understanding between TX and RX.

Our approach to semantic equalization exploits the presence of MIMO wireless channels. Thus, the first step is to build a complex semantic latent space that is more suitable for efficient MIMO communication. Assuming, without loss of generality, that d is even, this approach involves pairing the first half of the semantic features in $\mathbf{s}_T \in \mathbb{R}^d$ with the second half to form complex symbols, yielding an input vector $\mathbf{x} \in \mathbb{C}^{\frac{d}{2}}$. Then, assuming the TX is endowed with N_T antennas, we exploit a *semantic MIMO precoder* that maps the complex vector $\mathbf{x} \in \mathbb{C}^{\frac{d}{2}}$ into the vector $\bar{\mathbf{x}} \in \mathbb{C}^{KN_T}$ to be transmitted over K wireless channel usages. The semantic MIMO precoder implements a learnable transformation represented by the function $f : \mathbb{C}^{\frac{d}{2}} \rightarrow \mathbb{C}^{KN_T}$. This transformation additionally enables the semantic compression of the TX latent space (since we typically consider $\frac{d}{2} \gg KN_T$), with a compression factor defined as:

$$\zeta = \frac{KN_T}{d/2}. \quad (1)$$

We assume a MIMO flat Rayleigh fading channel described by the matrix $\bar{\mathbf{H}} \in \mathbb{C}^{N_T \times N_R}$, where N_R denotes the number of antennas at the RX side, and each element h_{ij} in $\bar{\mathbf{H}}$ is modeled as a zero-mean complex Gaussian random variable representing the fading effect between the i -th transmit antenna and the j -th receive antenna. We assume the channel to be constant within the time span by K consecutive transmissions. Then, at the RX side, we have a *semantic MIMO decoder* that maps the received symbols into a complex vector $\mathbf{y} \in \mathbb{C}^{\frac{m}{2}}$, via the learnable transformation $g : \mathbb{C}^{KN_R} \rightarrow \mathbb{C}^{\frac{m}{2}}$. Overall, the considered semantic MIMO communication channel can be compactly written as:

$$\hat{\mathbf{y}} = g(\mathbf{H}f(\mathbf{x}) + \mathbf{v}), \quad (2)$$

where $\mathbf{H} = \mathbf{I}_K \otimes \bar{\mathbf{H}} \in \mathbb{C}^{KN_R \times KN_T}$, with \mathbf{I}_K denoting the $K \times K$ identity matrix, and \otimes being the Kronecker product; also, $\mathbf{v} \in \mathbb{C}^{KN_R}$ represents the receiver noise vector, which follows a distribution $\mathcal{CN}(\mathbf{0}, \Sigma_v)$. Finally, the complex latent vector $\hat{\mathbf{y}} \in \mathbb{C}^{\frac{m}{2}}$ in (2) is converted into an m -dimensional real latent vector, say $\hat{\mathbf{s}}_R$, by inverting the halving operation done at the TX side. Our aim is to act on the learnable transformations f and g in (2) to perform the best possible alignment between TX and RX semantic latent spaces. This can be obtained by minimizing the (semantic) distance between the spaces

composed by the vectors \mathbf{s}_T and $\hat{\mathbf{s}}_R$ over a training set \mathcal{P} of available data points, which represent *semantic pilots* (SPs) in our context. In the sequel, we will illustrate two methods to perform and optimize semantic channel equalization based on the model (2), considering different complexities and imposing constraints on the TX power.

III. JOINT OPTIMIZATION OF SEMANTIC PRECODER AND DECODER FOR LATENT SPACE ALIGNMENT

In this section, we introduce two methods for joint optimization of semantic precoding/decoding to enable latent space alignment between semantic feature spaces at the TX and RX sides. The first one assumes linear precoding/decoding modules, whereas the second method exploits nonlinear precoding/decoding modules modeled by DNNs. In the sequel, we assume perfect estimation of the MIMO channel matrix \mathbf{H} ; the effect of imperfect channel state information (CSI) will be investigated in our future works.

A. Linear Semantic Precoding and Decoding

Here, we focus on the linear case, which provides a simplified yet effective approach for modeling semantic encoding and decoding functions, allowing for tractable optimization. Specifically, the semantic encoding function, f , is modeled as a linear transformation, represented by matrix $\mathbf{F} \in \mathbb{C}^{KN_T \times \frac{d}{2}}$, and the decoding function, g , is modeled by matrix $\mathbf{G} \in \mathbb{C}^{\frac{m}{2} \times KN_R}$. Here, w.l.o.g., we consider the application of *pre-whitening* to standardize the covariance of the transmitted symbols (\mathbf{x}) to the identity matrix. This assumption helps formulate a constraint on the TX power that is expressed solely in terms of \mathbf{F} , i.e., $\text{tr}(\mathbf{FF}^H) \leq P_T$, with $P_T > 0$ denoting the maximum power budget. Hence, (2) can be cast as:

$$\hat{\mathbf{y}} = \mathbf{GHF}\mathbf{x} + \mathbf{G}\mathbf{v}. \quad (3)$$

Our optimization criterion aims at minimizing the (semantic) distance between the signals given by model (3) and the target RX latent space, which is denoted by $\mathbf{y} \in \mathbb{C}^{\frac{m}{2}}$ (i.e., the complex version of the RX latent vectors associated with training data without considering channel impairments), with respect to linear semantic precoder and decoder. This facilitates the alignment of the latent spaces while introducing robustness against channel effects. While several distance metrics can be exploited in this formulation, in the sequel, we use mean-squared error as a simple yet effective way to measure latent space mismatches. In formulas, this translates to:

$$\begin{aligned} \min_{\mathbf{G}, \mathbf{F}} \quad & \mathbb{E} \|\mathbf{y} - (\mathbf{GHF}\mathbf{x} + \mathbf{G}\mathbf{v})\|_2^2 \\ \text{s.t.} \quad & \text{tr}(\mathbf{FF}^H) \leq P_T, \end{aligned} \quad (4)$$

where \mathbb{E} denotes the expected value over the noise and data distributions. Assuming to have $n = |\mathcal{P}|$ i.i.d. training samples $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^n$ and leveraging the zero-mean property of the AWGN, the objective function of (4) can be approximated using the empirical loss

$$\frac{1}{n} \sum_{i=1}^n \|\mathbf{y}^{(i)} - (\mathbf{GHF}\mathbf{x}^{(i)})\|_2^2 + \text{tr}(\mathbf{G}\Sigma_v\mathbf{G}^H), \quad (5)$$

with Σ_v denoting the noise covariance matrix. Let us define $\mathbf{X} \in \mathbb{C}^{\frac{d}{2} \times n}$ be the matrix that collects all training samples, where each column \mathbf{x}_i represents an individual input, and $\mathbf{Y} \in \mathbb{C}^{\frac{m}{2} \times n}$ be the matrix containing the corresponding receiver latent column vectors. The optimization problem (4) can be rewritten in matrix form as:

$$\begin{aligned} \min_{\mathbf{G}, \mathbf{F}} \quad & \frac{1}{n} \|\mathbf{Y} - \mathbf{GHFX}\|_F^2 + \text{tr}(\mathbf{G}\Sigma_v\mathbf{G}^H) \\ \text{s.t.} \quad & \text{tr}(\mathbf{FF}^H) \leq P_T. \end{aligned} \quad (6)$$

Given the bi-convex nature of problem (6), an efficient solution involves constructing an iterative algorithm that alternates between the solutions of \mathbf{G} and \mathbf{F} . To this aim, we reformulate the problem to be amenable for ADMM solution, which allows us to efficiently handle the constraints and solve the bi-convex optimization program in an efficient manner [33]. Then, let us define Ω as the set of elements that satisfy the given constraint:

$$\Omega = \{\mathbf{F} \mid \text{tr}(\mathbf{FF}^H) \leq P_T\}. \quad (7)$$

The indicator function $I_\Omega(\mathbf{F})$ for the set Ω is defined as:

$$I_\Omega(\mathbf{F}) = \begin{cases} +\infty & \text{if } \mathbf{F} \notin \Omega, \\ 0 & \text{if } \mathbf{F} \in \Omega. \end{cases} \quad (8)$$

Thus, using (8), and introducing an auxiliary variable $\mathbf{Z} = \mathbf{F}$, problem (4) can be equivalently rewritten as:

$$\begin{aligned} \min_{\mathbf{G}, \mathbf{F}, \mathbf{Z}} \quad & \frac{1}{n} \|\mathbf{Y} - \mathbf{GHFX}\|_F^2 + \text{tr}(\mathbf{G}\Sigma_v\mathbf{G}^H) + I_\Omega(\mathbf{Z}) \\ \text{s.t.} \quad & \mathbf{F} - \mathbf{Z} = \mathbf{0}. \end{aligned} \quad (9)$$

Problem (9) can now be solved using the scaled version of ADMM [33], which hinges on the following iterative updates:

$$\begin{aligned} \mathbf{G}^{(t+1)} = \arg \min_{\mathbf{G}} \quad & \frac{1}{n} \|\mathbf{Y} - \mathbf{GHF}^{(t)}\mathbf{X}\|_F^2 \\ & + \text{tr}(\mathbf{G}\Sigma_v\mathbf{G}^H) \end{aligned} \quad (10)$$

$$\begin{aligned} \mathbf{F}^{(t+1)} = \arg \min_{\mathbf{F}} \quad & \frac{1}{n} \|\mathbf{Y} - \mathbf{G}^{(t+1)}\mathbf{HF}\mathbf{X}\|_F^2 \\ & + \rho \|\mathbf{F} - \mathbf{Z}^{(t)} + \mathbf{U}^{(t)}\|_F^2 \end{aligned} \quad (11)$$

$$\mathbf{Z}^{(t+1)} = \text{Proj}_\Omega(\mathbf{F}^{(t+1)} + \mathbf{U}^{(t)}) \quad (12)$$

$$\mathbf{U}^{(t+1)} = \mathbf{U}^{(t)} + \mathbf{F}^{(t+1)} - \mathbf{Z}^{(t+1)} \quad (13)$$

where t denotes the iteration index, \mathbf{U} is the Lagrange multiplier used to enforce the (matrix) constraint in (9), ρ is a positive regularization parameter, and Proj_Ω denotes the orthogonal projection onto the set Ω in 7. The optimization procedure is initialized with $\mathbf{F}^{(0)} \sim \mathcal{CN}(0, 1)$, and both $\mathbf{Z}^{(0)}$ and $\mathbf{U}^{(0)}$ are set to zero matrices. In the sequel, we give closed-form expressions for the ADMM steps in (10)-(12).

The G-step. At the $(t+1)$ -th iteration, with \mathbf{F} fixed, the update for \mathbf{G} can be obtained by setting the gradient of (10) with respect to \mathbf{G}^H to zero. Recasting the objective of (10) as

$$\frac{\text{tr}((\mathbf{Y} - \mathbf{GHF}^{(t)}\mathbf{X})(\mathbf{Y} - \mathbf{GHF}^{(t)}\mathbf{X})^H)}{n} + \text{tr}(\mathbf{G}\Sigma_v\mathbf{G}^H), \quad (14)$$

simple algebra leads to the optimal update:

$$\mathbf{G}^{(t+1)} = \mathbf{Y}(\mathbf{HF}^{(t)}\mathbf{X})^H \left((\mathbf{HF}^{(t)}\mathbf{X})(\mathbf{HF}^{(t)}\mathbf{X})^H + n\Sigma_v \right)^{-1}. \quad (15)$$

The F-step. With \mathbf{G} , \mathbf{Z} and \mathbf{U} fixed, hinging again on expression (14), the update for \mathbf{F} at the $(t+1)$ -th iteration can be obtained by setting to zero the gradient of the objective function in (11) with respect to \mathbf{F}^H and solving it in closed form. The derivations lead to the following equation:

$$\begin{aligned} & (\mathbf{G}^{(t+1)}\mathbf{H})^H(\mathbf{G}^{(t+1)}\mathbf{H})\mathbf{F}(\mathbf{XX}^H) + n\rho\mathbf{F} \\ & - n\rho(\mathbf{Z}^{(t)} - \mathbf{U}^{(t)}) - (\mathbf{G}^{(t+1)}\mathbf{H})^H\mathbf{Y}\mathbf{X}^H = 0, \end{aligned} \quad (16)$$

Now, letting

$$\mathbf{A} = (\mathbf{G}^{(t+1)}\mathbf{H})^H(\mathbf{G}^{(t+1)}\mathbf{H}), \quad (17)$$

$$\mathbf{B} = \mathbf{XX}^H, \quad (18)$$

$$\mathbf{C} = n\rho(\mathbf{Z}^{(t)} - \mathbf{U}^{(t)}) + (\mathbf{G}^{(t+1)}\mathbf{H})^H\mathbf{Y}\mathbf{X}^H, \quad (19)$$

we can compactly write (16) as:

$$\mathbf{AFB} + n\rho\mathbf{F} = \mathbf{C}. \quad (20)$$

Then, exploiting $\text{vec}(\mathbf{AFB}) = (\mathbf{B}^H \otimes \mathbf{A})\text{vec}(\mathbf{F})$ in (20) and solving for $\text{vec}(\mathbf{F})$, we obtain

$$\text{vec}(\mathbf{F}) = (\mathbf{B}^H \otimes \mathbf{A} + n\rho\mathbf{I})^{-1}\text{vec}(\mathbf{C}),$$

where $\text{vec}(\cdot)$ denotes the vectorization operator. Finally, the update for \mathbf{F} at the $(t+1)$ -th step is:

$$\mathbf{F}^{(t+1)} = \text{vec}^{-1}((\mathbf{B}^H \otimes \mathbf{A} + n\rho\mathbf{I})^{-1}\text{vec}(\mathbf{C})). \quad (21)$$

with \mathbf{A} and \mathbf{C} depending on $\mathbf{G}^{(t+1)}$, $\mathbf{Z}^{(t)}$ and $\mathbf{U}^{(t)}$ as in (17)-(19). To avoid the unfavorable scaling of the Kronecker product for large matrices, (20) can be also recasted as a Sylvester equation, which can be efficiently solved using the Bartels-Stewart algorithm [34].

The Z-step. Applying the definition of projection operator, the step in (12) can be recast as:

$$\begin{aligned} \mathbf{Z}^{(t+1)} = \text{Proj}_\Omega(\mathbf{F}^{(t+1)} + \mathbf{U}^{(t)}) \\ = \arg \min_{\mathbf{Z}} \|\mathbf{F}^{(t+1)} + \mathbf{U}^{(t)} - \mathbf{Z}\|_F^2 \\ \text{s.t.} \quad \text{tr}(\mathbf{ZZ}^H) \leq P_T \end{aligned} \quad (22)$$

This problem is convex, and its optimal solution can be found by imposing the Karush-Kuhn-Tucker (KKT) conditions [35]:

$$\nabla_{\mathbf{Z}^H} \|\mathbf{F}^{(t+1)} + \mathbf{U}^{(t)} - \mathbf{Z}\|_F^2 + \lambda(\text{tr}(\mathbf{ZZ}^H) - P_T) = \mathbf{0} \quad (23)$$

$$\text{tr}(\mathbf{ZZ}^H) - P_T \leq 0$$

$$\lambda \geq 0$$

$$\lambda(\text{tr}(\mathbf{ZZ}^H) - P_T) = 0$$

Taking the gradient with respect to the conjugate transpose of the Lagrangian function in (23), we obtain:

$$\mathbf{Z} = \frac{1}{1+\lambda}(\mathbf{F}^{(t+1)} + \mathbf{U}^{(t)}). \quad (24)$$

Now, letting $\hat{\mathbf{Z}} = \mathbf{F}^{(t+1)} + \mathbf{U}^{(t)}$, and recalling that

$$\text{tr}(\mathbf{ZZ}^H) = \frac{1}{(1+\lambda)^2} \text{tr}(\hat{\mathbf{Z}}\hat{\mathbf{Z}}^H),$$

Algorithm 1 ADMM

- 1: **Input:** $\mathbf{F}^{(0)} \sim \mathcal{CN}(0, 1)$, $\mathbf{Z}^{(0)} = \mathbf{U}^{(0)} = \mathbf{0}$, n , Σ_v , and ρ .
 - 2: **Output:** Final values $\mathbf{F}^{(T)}$, $\mathbf{G}^{(T)}$, $\mathbf{Z}^{(T)}$, $\mathbf{U}^{(T)}$.
 - 3: **for** each iteration $t = 1, \dots, T$ **do**
 - 4: Update $\mathbf{G}^{(t+1)}$ as in (15).
 - 5: Update $\mathbf{F}^{(t+1)}$ as in (21).
 - 6: Update $\mathbf{Z}^{(t+1)}$ as in (26).
 - 7: Update $\mathbf{U}^{(t+1)}$ as in (13).
 - 8: **end for**
-

the optimal λ is given by:

$$\hat{\lambda} = \begin{cases} 0, & \text{if } \text{tr}(\hat{\mathbf{Z}}\hat{\mathbf{Z}}^H) \leq P_T, \\ \sqrt{\frac{\text{tr}(\hat{\mathbf{Z}}\hat{\mathbf{Z}}^H)}{P_T}} - 1, & \text{else.} \end{cases} \quad (25)$$

Thus, the update for \mathbf{Z} at the $(t + 1)$ -th step is:

$$\mathbf{Z}^{(t+1)} = \frac{1}{1 + \hat{\lambda}} (\mathbf{F}^{(t+1)} + \mathbf{U}^{(t)}). \quad (26)$$

The U-step. Finally, the update for \mathbf{U} at the $(t + 1)$ -th step is given by the recursive expression in (13).

The linear model in (3) offers simplicity, low complexity, and a robust training procedure based on ADMM. However, general nonlinear models can typically outperform linear ones, with a price paid in terms of complexity and interpretability. In the next paragraph, we introduce a nonlinear MIMO semantic precoding and decoding strategy based on DNNs.

B. Neural Semantic Precoding and Decoding

In this section, we introduce the neural network-based approach. Unlike the linear case, the semantic encoding and decoding functions are modeled as nonlinear parametric functions, such as DNNs. Specifically, we have a semantic neural precoder $f_\theta(\cdot)$, and a semantic neural decoder $g_\psi(\cdot)$, with a set of learnable and complex parameters θ and ψ , respectively. In this setting, we have a general observation model as in (2). The structure of the DNNs can be chosen according to different design choices, and the achievable performance depends on the specific latent spaces to be aligned, and the complexity constraints on the overall architecture. Furthermore, due to the specific complex nature of the signal, we adopt activation functions that operate on complex vectors. In particular, we exploit the phase-amplitude method proposed in [36], where a generic activation function $\alpha(\cdot)$ is applied to the magnitude of the complex number, while the phase remains unchanged. Specifically, for a general complex input $\mathbf{z} \in \mathbb{C}^k$, the activation function is defined as $\varphi(\mathbf{z}) = \alpha(|\mathbf{z}|) \cdot e^{j\phi}$, where $\phi = \tan^{-1}(\frac{b}{a})$ represents the phase of the complex vector $\mathbf{z} = \mathbf{a} + j\mathbf{b}$, with $\mathbf{a}, \mathbf{b} \in \mathbb{R}^k$. Finally, the parameters of the neural semantic precoders and decoders are optimized according to the following training criterion:

$$\min_{\theta, \psi} \mathbb{E} \|\mathbf{y} - (g_\psi(\mathbf{H}f_\theta(\mathbf{x}) + \mathbf{v}))\|^2 + \beta \|\theta\|_0 + \gamma \|\psi\|_0. \quad (27)$$

with $\beta, \gamma \geq 0$, where the regularization terms $\beta \|\theta\|_0$ and $\gamma \|\psi\|_0$ are used to promote sparsity of the model parameters at

Algorithm 2 PGD with Hard Thresholding

- 1: **Input:** Initial weights $\mathbf{w}^{(0)}$ and $\{\tau_i\}_{i=1}^{|\mathbf{w}|}$.
 - 2: **Output:** The final sparse weights $\mathbf{w}^{(T)}$.
 - 3: **for** each epoch $t \in \{1, \dots, T\}$ **do**
 - 4: Gradient descent update and global hard thresholding:

$$\mathbf{w}^{(t+1)} = \mathcal{H}_\tau(\mathbf{w}^{(t)} - \eta \nabla \mathcal{L}(\mathbf{w}^{(t)}));$$
 - 5: Freeze gradients of pruned weights:

$$\nabla \mathcal{L}(w_i) = 0, \quad \text{if } w_i^{(t+1)} = 0;$$
 - 6: **end for**
-

the TX and RX side, which reduces memory requirements for model storage and computational complexity at inference time. The presence of noise \mathbf{v} is addressed directly during the forward pass by adding random samples drawn from $\mathcal{CN}(\mathbf{0}, \Sigma_v)$ to the receiver signal $\mathbf{H}f_\theta(\mathbf{x})$. This makes the model naturally robust to the noise introduced by the communication channel. The transmitting power constraint is incorporated directly into the neural network structure by implementing ℓ_2 normalization of the semantic encoder output.

We solve problem (27) using a Proximal Gradient Descent (PGD) [37] method with Hard Thresholding, which is described in Algorithm 2. Specifically, after each training epoch, a hard-thresholding operator $\mathcal{H}_\tau(\cdot)$ is applied to the stacked weight vector $\mathbf{w} := [\theta, \psi]$ in the following manner:

$$\mathcal{H}_\tau(w_i) = \begin{cases} w_i, & \text{if } |w_i| > \tau_i, \\ 0, & \text{otherwise.} \end{cases} \quad (28)$$

where $\{\tau_i\}_{i=1}^{|\mathbf{w}|}$ is a set of thresholds where each element is defined as:

$$\tau_i = \begin{cases} \beta \cdot \eta, & \text{if } \mathbf{w}_i \in \theta; \\ \gamma \cdot \eta, & \text{if } \mathbf{w}_i \in \psi. \end{cases} \quad (29)$$

with η being the learning rate.

C. Computational Complexity Analysis

In this section, we measure the complexity of the models in terms of the number of floating-point operations (FLOPs) required for computations. We consider each arithmetic operation to count as one FLOP.

Precisely, to compute the multiplication of a matrix $\mathbf{A} \in \mathbb{R}^{a \times b}$ and a vector $\mathbf{b} \in \mathbb{R}^b$, $a(2b - 1)$ FLOPs are required. When we consider the bias component, which involves adding a vector $\mathbf{c} \in \mathbb{R}^a$, the FLOPs increase by a , resulting in a total $2ab$ FLOPs. When \mathbf{A} , \mathbf{b} , and \mathbf{c} are complex, the required FLOPs are $a(8b - 2)$ for the matrix-vector multiplication and $8ab$ considering the bias too. We explicitly incorporate matrix sparsity into our DNN models to reduce computational requirements and improve storage efficiency. The sparsity level is defined as $s \in [0, 1]$, where s represents the fraction of zero elements in the matrix \mathbf{A} . Thus, the total number of non-zero elements in \mathbf{A} is assumed to be $(1 - s)ab$. Consequently, the number of FLOPs required for a forward pass through a complex linear layer can be approximated as $8(1 - s)ab$. This formulation demonstrates how sparsity

TABLE I: Approximated FLOP counts for the considered cases.

Case	Complex	Sparse	Dense	FLOPs
Ab	✓		✓	$a(8b - 2)$
Ab + c	✓		✓	$8ab$
Ab + c	✓	✓		$8(1 - s)ab$

effectively reduces computational costs. For the sake of clarity, a summary of FLOPs calculations is provided in Table I.

Linear model FLOPs. Consisting of the implementation of two complex dense matrices $\mathbf{F} \in \mathbb{C}^{KN_T \times \frac{d}{2}}$ and $\mathbf{G} \in \mathbb{C}^{\frac{m}{2} \times KN_R}$, the total number of FLOPs required for the linear model is:

$$\begin{aligned} \#\text{FLOPs} &= KN_T \left(8\frac{d}{2} - 2 \right) + \frac{m}{2} (8KN_R - 2) \\ &= 4KN_T d + (4KN_R - 1)m - 2KN_T \end{aligned} \quad (30)$$

Neural model FLOPs. Let i_p , o_p , h_p , and l_p be the number of neurons in the input layer, output layer, hidden layers, and the number of hidden layers for the precoder, respectively. Similarly, we have i_d , o_d , h_d , and l_d for the decoder side. Then, given a level of sparsity s , the total number of FLOPs required by the neural network solution, under the assumption of uniform sparsity in the weights, is given by:

$$\begin{aligned} \#\text{FLOPs} &= 8(1 - s)[h_p i_p + o_p h_p + h_d i_d + o_d h_d + (l_p - 1)h_p^2 \\ &\quad + (l_d - 1)h_d^2] + c(l_p h_p + l_d h_d) \end{aligned} \quad (31)$$

where c represents the FLOPs required by the complex activation function $\varphi(\cdot)$.

IV. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed semantic equalization methods through numerical experiments. We used the CIFAR-10 dataset [38], comprising 60,000 32×32 color images distributed across 10 classes. Among them, 42,500 images were used for training, from which we uniformly sample three subsets with balanced class distributions, resulting in a total of $n \in \{420, 4200, 42000\}$ SPs. The remaining 7500 images are used for validation, and 10000 for testing, with classification across 10 labels as the downstream task, \mathcal{T} . Furthermore, we employed two encoding models from the Python library *timm* [39]: (i) The *vit_small_patch16_224* model to generate the transmitter-side encodings with $d = 384$; (ii) the *vit_base_patch16_224* model for the receiver-side encodings with $m = 768$. For simplicity, we consider square MIMO Rayleigh fading channels with unitary variance, where the number of transmitting and receiving antennas are equal, i.e., $N_T = N_R$. All results were averaged across six random seeds, considering $\rho = 100$.¹

The neural semantic precoder and decoder are implemented using a multilayer perceptron architecture, each with a single hidden layer ($l_p = l_d = 1$), consisting of $h_p = \frac{d}{2} = 192$ and $h_d = \frac{m}{2} = 384$ neurons, respectively. From (31), the FLOPs complexity reduces to $\#\text{FLOPs} = 2(1 - s)[d^2 + m^2 + 2K(N_T d + N_R m)] + \frac{c}{2}(d + m)$. The linear model was trained for 20 iterations, while the neural network was trained for 50

epochs with a learning rate of $\eta = 10^{-3}$. In the neural model, we set $\beta = \gamma$ to set a global threshold τ .

Baselines. We compare our methodology with baselines that directly transmit vectors \mathbf{x} of the TX latent space while performing semantic alignment and MIMO channel equalization in a disjoint fashion. Specifically, letting $\bar{\mathbf{H}} = \mathbf{U}\Sigma\mathbf{V}^H$, the baselines involve SVD-based MIMO channel equalization, where precoding and decoding are chosen as:

$$\mathbf{F} = \mathbf{1}_K \otimes \mathbf{V}, \quad (32)$$

$$\mathbf{G} = \mathbf{1}_K^T \otimes \left(\Sigma^H \Sigma + \frac{1}{\text{SNR}} \mathbf{I}_{N_R} \right)^{-1} (\mathbf{U}\Sigma)^H, \quad (33)$$

where $\mathbf{1}_K$ is the K -dimensional unitary vector, and SNR represents the Signal-to-Noise Ratio (SNR) of the communication channel. Semantic alignment is performed by mapping the TX latent space into the RX one via a linear transformation obtained by solving the following least-square problem:

$$\min_{\mathbf{Q} \in \mathbb{R}^{m \times d}} \frac{1}{n} \sum_{i=1}^n \|\mathbf{s}_R^{(i)} - \mathbf{Q}\mathbf{s}_T^{(i)}\|_2^2. \quad (34)$$

To assess the performance of the baselines for different amounts of transmitted information, we consider three alternative strategies: (i) the First- κ baseline (First- κ), which transmits the first $2KN_T$ features of \mathbf{s}_T ; (ii) the Top- κ baseline (Top- κ), which selects and transmits the largest KN_T features of \mathbf{s}_T in terms of ℓ_1 -norm; and (iii) the Eigen- κ baseline (Eigen- κ) in which the TX latent space is first precoded using $\tilde{\mathbf{F}} = \tilde{\Sigma}_Q V_Q^T$ and then after MIMO equalization via SVD the RX latent space is decoded using $\tilde{\mathbf{G}} = U_Q \tilde{\Sigma}_Q$, where $U_Q \Sigma_Q V_Q^T$ is the singular value decomposition of the matrix \mathbf{Q} in (34), and $\tilde{\Sigma}_Q$ is a filtered version of the Σ_Q in which only the largest $2KN_T$ eigenvalues are retained. For First- κ and Top- κ baselines, the alignment is performed once the TX latent vectors are received by RX. Specifically, for Top- κ baseline, to ensure accurate signal reconstruction at the receiver, the transmitted signal is structured such that half of the transmission space is allocated to the indices corresponding to the transmitted features, while the remaining half is dedicated to the features themselves. This allocation effectively doubles the communication burden. It is assumed that the indices are perfectly reconstructed at the receiver. Given d , and choosing K and N_T , we can set a specific compression ratio in (1) for the three baselines.

Comparisons. We illustrate the performance of our approach in Fig. 2, which shows the average accuracy of the classification task as a function of the compression factor ζ , for varying numbers of SPs. Specifically, we compare: (i) the proposed neural semantic precoding/decoding strategy in Sec. III.B; (ii) The linear semantic precoding/decoding method introduced in Sec. III.A; and (iii) the Eigen- κ baseline. For the proposed neural and linear methods, we also examine their optimization in the absence of wireless channel knowledge, referring to these strategies as “channel-unaware.” The results are obtained considering an SNR equal to 20 dB, and setting $\beta = \gamma = 0$. We explore different compression factors ζ by setting $K = 1$ and varying the number of antennas. As we can notice from Fig. 2, the proposed semantic equalization

¹<https://github.com/SPAICom/semantic-alignment-mimo.git>

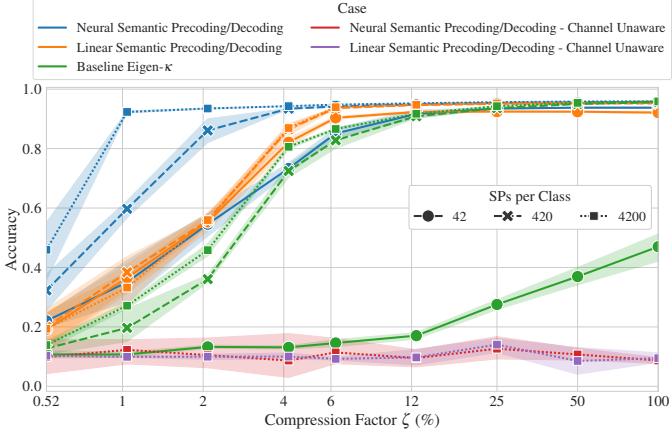


Fig. 2: Accuracy versus ζ , with $\text{SNR} = 20 \text{ dB}$ and $\beta = \gamma = 0$.

strategies outperform the Eigen- κ baseline, showing excellent performance even at large compression levels. For instance, the proposed neural approach achieves nearly 90% accuracy while transmitting only two complex symbols through the channel, i.e., $\zeta \approx 1\%$ of the original dimension in terms of transmitted symbols. Moreover, at small compression factors, the linear method outperforms the Eigen- κ baseline by roughly ten percentage points, demonstrating the advantages of joint over disjoint design. Comparable performance is observed after $\zeta \approx 12\%$, whereas the linear model achieves peak performance at $\zeta \approx 6\%$. Unsurprisingly, the proposed neural method is more affected by the number of SPs used for alignment, while the linear counterpart remains relatively robust. In contrast, the baseline performs poorly when only a limited number of SPs are available. As expected, the channel-unaware methods perform poorly, further reinforcing the importance of incorporating physical channel information for effective semantic channel equalization in wireless communications.

As a further result, in Fig. 3, we illustrate the behaviour of the average accuracy versus the SNR, comparing the proposed strategies with all the baselines. In this simulation, we consider $K=1$, 4,200 SPs per class and $\zeta \approx 4\%$. As expected, the accuracy increases at larger SNR, with the neural approach consistently outperforming the linear counterpart. The performance gap previously observed between the linear method and the Eigen- κ baseline remains evident across different SNR levels, as illustrated in Fig. 3. This consistency suggests that joint optimization not only improves performance under high compression but also enhances robustness to noise, likely by leveraging the channel structure during semantic compression. Notably, the neural model delivers excellent performance even considering a limited number of antennas, only one channel usage, and low SNR levels. However, this performance improvement comes with a price in terms of computational complexity. As an example, the neural model requires approximately 113 times more FLOPs than the linear model for $\zeta \approx 3\%$. Nevertheless, NNs are typically overparametrized for the task at hand, and it becomes interesting to investigate the performance-complexity trade-off achievable through sparsification of the NN weights. To this aim, we train the neural model as in (27) using Algorithm 1, varying the hyper-

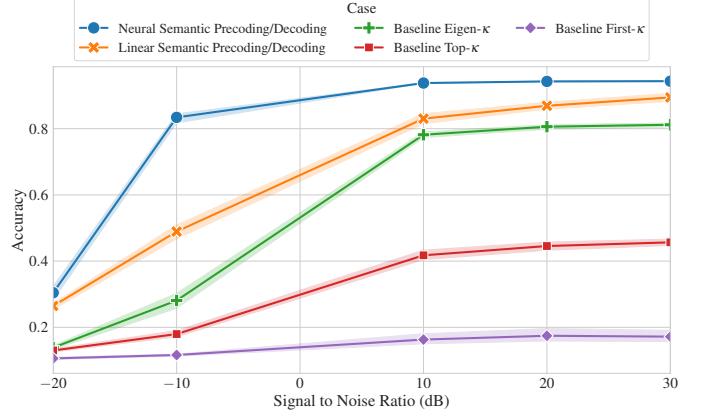


Fig. 3: Accuracy versus SNR, with $\zeta \approx 4\%$ and 4,200 SPs per class.

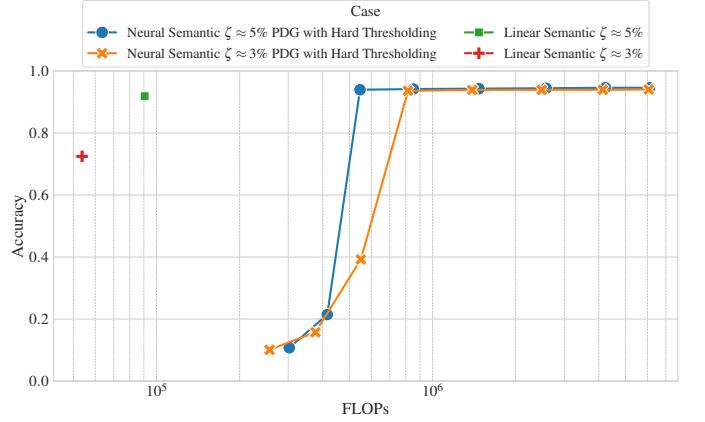


Fig. 4: Accuracy versus FLOPs, with $\zeta \approx 3\%$ and $\zeta \approx 5\%$, $\text{SNR} = 20 \text{ dB}$ and 4,200 SPs per class.

parameters $\beta = \gamma$ across the set $\{0, 10, 20, 30, 40, 50, 60, 70\}$ to enforce different level of weight sparsity and, consequently, different computational complexity (cf. 31). The results are reported in Fig. 4, which illustrates the behavior of the average accuracy versus the number of FLOPs, comparing the linear and neural approaches, for different compression factors, with $\text{SNR} = 20 \text{ dB}$ and 4,200 SPs per class. However, as the number of FLOPs approaches that of the linear model, performance degrades significantly, making the linear method optimal under tight budgets, while the neural model is preferable when resources permit.

V. CONCLUSIONS

This paper introduces a novel framework for latent space alignment in MIMO semantic communications, addressing the challenges posed by both semantic mismatch and channel distortions. Our approach integrates semantic compression with the alignment of semantic spaces through a joint optimization of MIMO semantic precoding and decoding, ensuring robust and efficient transmission. To solve this problem, we propose and evaluate two distinct methodologies. The first is a linear optimization approach, formulated as a biconvex optimization problem and solved using ADMM. The second

is a neural network-based solution that learns an optimized mapping between semantic representations and the transmitted signals. Numerical results validate the proposed methodologies. Furthermore, the neural model consistently outperforms the linear one across various channel conditions and noise levels, achieving superior alignment and robustness. However, this performance gain comes at the cost of increased computational complexity. To mitigate this issue, we applied sparsification to the neural network model using Proximal Gradient Descent with Hard Thresholding, significantly reducing the FLOPs while preserving competitive accuracy. However, as the number of FLOPs approaches that of the linear model, performance degrades significantly. This finding highlights the suitability of the linear approach in scenarios with stringent computational limitations, while the neural model emerges as the better choice when computational resources allow for greater flexibility and performance is the primary concern. Future work will focus on further optimizations, leveraging the dynamic and reconfigurable nature of the channel matrix to improve adaptability, or considering more challenging multi-user semantic communication scenarios.

REFERENCES

- [1] C. De Alwis, A. Kalla, Quoc-Viet Pham, P. Kumar, K. Dev, Won-Joo Hwang, and M. Liyanage, “Survey on 6g frontiers: Trends, applications, requirements, technologies and future research,” *IEEE Open Journal of the Communications Society*, vol. 2, pp. 836–886, 2021.
- [2] G. Shi, D. Gao, X. Song, J. Chai, M. Yang, X. Xie, L. Li, and X. Li, “A new communication paradigm: From bit accuracy to semantic fidelity,” *arXiv preprint arXiv:2101.12649*, 2021.
- [3] R. Li, “Towards a new internet for the year 2030 and beyond,” *Proc. 3rd Annu. ITU IMT-2020/5G Workshop Demo Day*, pp. 1–21, 2018.
- [4] E. Calvanese Strinati and S. Barbarossa, “6G networks: Beyond shannon towards semantic and goal-oriented communications,” *Computer Networks*, vol. 190, pp. 107930, 2021.
- [5] T. Getu, G. Kaddoum, and M. Bennis, “Semantic communication: A survey on research landscape, challenges, and future directions,” *Authorea Preprints*, 2023.
- [6] E. Calvanese Strinati, P. Di Lorenzo, et al., “Goal-oriented and semantic communication in 6G AI-native networks: The 6G-goals approach,” *arXiv preprint arXiv:2402.07573*, 2024.
- [7] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, Kai Kit Wong, and Chan-Byoung Chae, “Beyond transmitting bits: Context, semantics, and task-oriented communications,” *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 5–41, 2022.
- [8] X. Luo, Hsiao-Hwa Chen, and Q. Guo, “Semantic communications: Overview, open issues, and future research directions,” *IEEE Wireless Communications*, vol. 29, no. 1, pp. 210–219, 2022.
- [9] B. Güler, A. Yener, and A. Swami, “The semantic communication game,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 787–802, 2018.
- [10] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, “Towards a theory of semantic communication,” in *2011 IEEE Network Science Workshop*. IEEE, 2011, pp. 110–117.
- [11] M. Kountouris and N. Pappas, “Semantics-empowered communication for networked intelligent systems,” *IEEE Communications Magazine*, vol. 59, no. 6, pp. 96–102, 2021.
- [12] P. A. Stavrou and M. Kountouris, “The role of fidelity in goal-oriented semantic communication: A rate distortion approach,” *IEEE Transactions on Communications*, 2023.
- [13] F. Binucci, P. Banelli, P. Di Lorenzo, and S. Barbarossa, “Adaptive resource optimization for edge inference with goal-oriented communications,” *EURASIP J. on Advances in Signal Proc.*, , no. 1, pp. 123, 2022.
- [14] P. Di Lorenzo, M. Merluzzi, F. Binucci, C. Battiloro, P. Banelli, E. Calvanese Strinati, and S. Barbarossa, “Goal-oriented communications for the IoT: System design and adaptive resource optimization,” *IEEE Internet of Things Magazine*, vol. 6, no. 4, pp. 26–32, 2023.
- [15] S. Fiorellino, C. Battiloro, E. Calvanese Strinati, and P. Di Lorenzo, “Dynamic relative representations for goal-oriented semantic communications,” *arXiv preprint arXiv:2403.16986*, 2024.
- [16] C. Kurisummoottil Thomas, E. Calvanese Strinati, and W. Saad, “Reasoning with the theory of mind for pragmatic semantic communication,” *arXiv preprint arXiv:2311.18224*, 2023.
- [17] S. Barbarossa, D. Comminiello, E. Grassucci, F. Pezone, S. Sardellitti, and P. Di Lorenzo, “Semantic communications based on adaptive generative models and information bottleneck,” *IEEE Communications Magazine*, vol. 61, no. 11, pp. 36–41, 2023.
- [18] H. Xie, Z. Qin, G. Ye Li, and Biing-Hwang Juang, “Deep learning enabled semantic communication systems,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.
- [19] C. E. Shannon, “A mathematical theory of communication,” *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [20] E. Bourtsoulatz, D. Burth 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.
- [21] M. Yang, C. Bian, and Hun-Seok Kim, “Ofdm-guided deep joint source channel coding for wireless multipath fading channels,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 584–599, 2022.
- [22] J. Xu, B. Ai, W. Chen, A. Yang, P. Sun, and M. Rodrigues, “Wireless image transmission using deep source channel coding with attention modules,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2315–2328, 2022.
- [23] J. Dai, S. Wang, K. Tan, Z. Si, X. Qin, K. Niu, and P. Zhang, “Nonlinear transform source-channel coding for semantic communications,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 8, pp. 2300–2316, 2022.
- [24] H. Wu, Y. Shao, C. Bian, K. Mikolajczyk, and D. Gündüz, “Deep joint source-channel coding for adaptive image transmission over mimo channels,” *IEEE Transactions on Wireless Communications*, vol. 23, no. 10, pp. 15002–15017, 2024.
- [25] J. Xu, Tze-Yang Tung, B. Ai, W. Chen, Y. Sun, and D. Gündüz, “Deep joint source-channel coding for semantic communications,” *IEEE Communications Magazine*, vol. 61, no. 11, pp. 42–48, 2023.
- [26] D. Alvarez-Melis, S. Jegelka, and T. S Jaakkola, “Towards optimal transport with global invariances,” in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1870–1879.
- [27] L. Moschella, V. Maiorca, M. Fumero, A. Norelli, F. Locatello, and E. Rodolà, “Relative representations enable zero-shot latent space communication,” *arXiv preprint arXiv:2209.15430*, 2022.
- [28] Z. Pan, J. Cai, and B. Zhuang, “Stitchable neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16102–16112.
- [29] M. Sana and E. Calvanese Strinati, “Semantic channel equalizer: Modelling language mismatch in multi-user semantic communications,” *arXiv preprint arXiv:2308.03789*, 2023.
- [30] Z. Lähner and M. Moeller, “On the direct alignment of latent spaces,” in *Proceedings of UniReps: the First Workshop on Unifying Representations in Neural Models*. PMLR, 2024, pp. 158–169.
- [31] V. Maiorca, L. Moschella, M. Fumero, F. Locatello, and E. Rodolà, “Latent space translation via inverse relative projection,” *arXiv preprint arXiv:2406.15057*, 2024.
- [32] T. Hüttébräcker, S. Fiorellino, M. Sana, P. Di Lorenzo, and E. Calvanese Strinati, “Relative representations of latent spaces enable efficient semantic channel equalization,” in *GLOBECOM 2024-2024 IEEE Global Communications Conference*. IEEE, 2024.
- [33] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [34] R. H. Bartels and G. W. Stewart, “Algorithm 432 [c2]: Solution of the matrix equation $ax + xb = c$ [f4],” *Communications of the ACM*, vol. 15, pp. 820 – 826, 1972.
- [35] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, “Kkt conditions, first-order and second-order optimization, and distributed optimization: Tutorial and survey,” 2021.
- [36] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. Felipe Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, “Deep complex networks,” 2018.
- [37] N. Parikh, S. Boyd, et al., “Proximal algorithms,” *Foundations and trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
- [38] A. Krizhevsky, G. Hinton, et al., “Learning multiple layers of features from tiny images,” 2009.
- [39] R. Wightman, “Pytorch image models,” <https://github.com/rwightman/pytorch-image-models>, 2019.