Received 4 September 2025; accepted 24 September 2025. Date of publication 2 October 2025; date of current version 13 October 2025.

Digital Object Identifier 10.1109/OJCOMS.2025.3617156

Semantic Communication for Cooperative Multi-Tasking Over Rate-Limited Wireless Channels With Implicit Optimal Prior

AHMAD HALIMI RAZLIGHI (Graduate Student Member, IEEE), CARSTEN BOCKELMANN (Member, IEEE), AND ARMIN DEKORSY (Senior Member, IEEE)

Department of Communications Engineering, University of Bremen, 28359 Bremen, Germany

CORRESPONDING AUTHOR: A. HALIMI RAZLIGHI (e-mail: halimi@ant.uni-bremen.de)

This work was supported by the German Federal Ministry of Research, Technology, and Space (BMFTR) under Grant 16KISK016 (Open6GHub).

ABSTRACT In this work, we expand the cooperative multi-task semantic communication framework (CMT-SemCom) introduced in [1], which divides the semantic encoder on the transmitter side into a common unit (CU) and multiple specific units (SUs), to a more applicable design. Our proposed system model addresses real-world constraints by introducing a general design that operates over rate-limited wireless channels. Further, we aim to tackle the rate-limit constraint, represented through the Kullback-Leibler (KL) divergence, by employing the density ratio trick alongside the implicit optimal prior method (IoPm). By applying the IoPm to our multi-task processing framework, we propose a hybrid-learning approach that combines deep neural networks with kernelized-parametric machine learning methods, enabling a robust solution for the CMT-SemCom. Our framework is grounded in information-theoretic principles and employs variational approximations to bridge theoretical foundations with practical implementations. Simulation results demonstrate the proposed system's effectiveness in rate-constrained multi-task SemCom scenarios, highlighting its potential for enabling intelligence in next-generation wireless networks.

INDEX TERMS Cooperative multi-tasking, deep learning, hybrid learning, information theory, implicit optimal prior, parametric methods, semantic communication.

I. INTRODUCTION

RECENT advancements in artificial intelligence, particularly in deep learning (DL) and end-to-end (E2E) communication technologies, have led to the rise of *semantic communication* (SemCom) [2], [3], [4], [5]. It has attracted significant attention, being recognized as a critical enabler for the sixth generation (6G) of wireless communication networks. SemCom is expected to play a key role in supporting a wide range of innovative applications that will define 6G connectivity and beyond [6]. This is because emerging applications often have to prioritize task execution over the precise reconstruction of transmitted information at the receiver.

In contrast to conventional communication systems, which are grounded in Shannon's information theory [7] and focus on the accurate transmission of symbols, SemCom prioritizes understanding the meaning and goals behind transmitted information. Therefore, designing appropriate communication systems requires moving beyond the traditional focus on precise bit transmission and rethinking the aspects that address communication problems. According to Shannon and Weaver's work, the communication problem is categorized into three levels, each addressing a specific issue [8]:

- The technical problem: Accurate transmission of symbols,
- The semantic problem: Transmitting the desired meaning precisely through symbols,
- The effectiveness problem: Effectiveness of the received meaning.

To meet the demands of emerging applications, SemCom operates at the second level of communication where the goal is to convey the desired meaning rather than ensuring exact bit-level accuracy. By surpassing the traditional focus on the

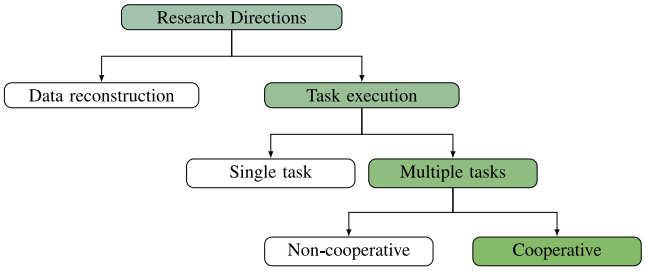


FIGURE 1. Categorization of research works in semantic communications.

precise transmission of bits, SemCom is well-suited for new applications, such as the industrial Internet and autonomous systems, where successful task execution is prioritized over the exact reconstruction of transmitted data at the receiver.

Research into SemCom has explored five main approaches, with four detailed in [9] and a fifth inspired by Weaver's extension of Shannon's theory to include the semantic level [10]. These approaches are:

- Classical approach,
- Knowledge graph approach,
- Machine learning (ML) approach,
- Significance approach,
- Information theory approach.

The classical approach utilizes *logical* probability to quantify semantic information. Bar-Hillel and Carnap [11], introduced this approach and have inspired many other works introducing methods to measure the semantic information of a source. As noted in [9], this definition of semantic information primarily applies to psychological investigations rather than communication counterparts.

Next, the knowledge graph approach represents semantics by knowledge graph structures. This approach stores the information such that the semantic relations between entities are held via semantic matching models as a knowledge graph technique [12]. For instance, [13] exploits this approach for its proposed semantic information detection framework, using triplets of the graph as semantic symbols.

The ML approach leverages learned model parameters to represent semantics. The ML approach lacks the communication-theoretic analysis in the semantic communication domain, relying on defined loss functions and the closed-box nature of its tools, such as deep neural networks (DNNs).

The significance approach considers the significance of information as its semantics. Although it is argued in [9] that this approach is more about investigating the effectiveness problem of communication, its application for the semantic problem has been studied emphasizing *timing* as semantics. This is specifically explored in the semantic communication domain under the Age of Information (AoI) topic [2].

Lastly, inspired by Weaver, an alternative approach extends Shannon's *statistical* probability (information theory) beyond the technical layer to the next two levels. Recently, some works have adopted information theory in investigating semantic communication, which are mentioned later.

The research work in SemCom primarily focuses on two research directions: data reconstruction and task execution, illustrated in Fig. 1. Initial investigations into data recovery were led by [14] and [15], which utilized the ML approach to reconstruct diverse data sources such as text, speech, and images. Building on these foundational works, [16], [17], [18], [19], [20] have extended the focus to explore communication concepts like efficiency and resource allocation in SemCom. In addition, SemCom systems dealing with structured data have been examined through the knowledge graph approach to enhance data recovery [21]. Recent developments also address robustness and reliability, for instance, [22] proposed a multi-functional reconfigurable intelligent surface-assisted framework for semantic anti-jamming communication, showing how semantic transceivers and RIS can jointly provide resilience.

On the other hand, task-oriented communication or goal-oriented communication can be categorized into single-task processing and multi-task processing. The latter is further divided into two directions: non-cooperative processing and cooperative multi-task processing. Our paper specifically addresses cooperative multi-task processing within the context of SemCom. A review of the literature related to task execution SemCom is provided in Section I-A.

To better situate our contribution, we note that there exist generally two main paradigms in multi-task learning [23]: (i) multiple tasks on different datasets, and (ii) multi-label tasks where different classification tasks are supported based on a single dataset. Our focus falls into the second category, focusing on a multi-label domain, where multiple semantic variables are extracted from the same observation and represent different classification tasks. This perspective is elaborated in detail in Section II.

A. RELATED WORKS

In task-oriented SemCom, the focus shifts to executing intelligent tasks at the receivers. Most research in this area has concentrated on single-task scenarios. For example, [24] developed a communication scheme using the information bottleneck (IB) framework, which encodes information while adapting to dynamic channel conditions. Moreover, the same authors in [25] studied distributed relevant information encoding for collaborative feature extraction to fulfill a single task. Reference [26] also offered a framework for collaborative retrieval of the message using multiple received semantic information. Recent studies have highlighted the integration of communication with computation and sensing (ISCC) in this context. In particular, [27] investigated a multi-device edge inference with ISCC for improved inference accuracy in a classification task.

To address practical communication scenarios, SemCom systems must be capable of handling multiple tasks simultaneously. Early efforts, such as [28], [29], explored non-cooperative methods where each task operates on its

respective dataset independently. Conversely, recent works like [30], [31], [32] studied joint multi-tasking using established ML approaches and architecture [33] for SemCom systems. Although these works incorporated communication aspects like channel conditions in their studies, their multi-task processing is based exclusively on ML approaches.

On the other hand, in [1], we introduced an information-theoretic analysis of a cooperative multi-task (CMT) SemCom system, avoiding the closed-box use of DNN. Reference [1] investigated a split structure for the semantic encoder, dividing the semantic encoder into a common unit (CU) and multiple specific units (SUs), to enable cooperative processing of various tasks on the transmitter side. The proposed CMT-SemCom can perform multi-tasking based on a single observation. Further, [34] expanded the CMT-SemCom to scenarios, in which, instead of full observation, distributed partial observations are available. By introducing CCMT-SemCom for multi-tasking in [34], we combined the cooperative processing on the transmitter side with the collaborative processing, where multiple nodes collaborate to execute their shared task, on the receive side. In addition, on exploring the physical layer communications aspects, [35] has studied resource allocation for multi-task SemCom networks.

TABLE 1. The table of notations.

Notation	Definition
$\mathbf{S}$	observation (input)
$\mathbf{z}$	semantic variables
$\mathbf{c}$	output of the CU
$\mathbf{x}_n$	output of the n-th SU encoder
$\mathbf{n}$	additive white Gaussian noise
$\hat{\mathbf{x}}_n$	noise-corrupted version of $\mathbf{x}_n$
$I(\cdot;\cdot)$	mutual information
$KL(\cdot \parallel \cdot)$	Kullback-Leibler divergence
$\mathbb{E}[\cdot]$	expectation
$\mathcal{L}(\cdot)$	objective function
$\boldsymbol{\theta}$	neural network (NN) parameters of the CU encoder
$\boldsymbol{\Xi}$	NN parameters of the auxiliary CU decoders
$\boldsymbol{\phi}_n$	NN parameters of the n-th SU encoder
$\boldsymbol{\psi}_n$	NN parameters of the n-th SU decoder
$\boldsymbol{\mu}$, $\boldsymbol{\sigma}$	mean and standard deviation of a Gaussian distribution
ϵ	auxiliary random variable for reparameterization trick
$r(\cdot)$	density ratio function
$\boldsymbol{\omega}$	parameter vector of the density ratio estimator
$\Omega(\cdot)$	basis function
$K(\cdot,\cdot)$	kernel function
σ_k	kernel bandwidth

B. MOTIVATIONS AND CONTRIBUTIONS

This work builds upon the CMT-SemCom framework introduced in [1] and extends it to a more realistic setting by incorporating rate-limited wireless communication channels. The presence of this constraint introduces a Kullback-Leibler (KL) divergence term in the objective function of the specific units, which must be handled during the learning step.

To better address this constraint, we propose a separation-based design where the CU and the SUs are optimized in turn. This not only clarifies their distinct functional roles but also leads to a more tractable formulation of the constrained learning problem. In addition, as shown in recent research works, i.e., [16], [34], such a separation-based design offers better compatibility with handling different channel conditions by reducing the number of trained parameters for each channel condition.

Existing approaches rely on a fixed prior when regularizing the KL term, e.g., [24], [36], which can limit the flexibility and performance of the system. In contrast, we propose to adopt the Implicit Optimal Prior method (IoPm) in this work, which leverages density ratio estimation to better approximate the prior in a data-driven manner. However, while investigating, we found that directly integrating IoPm into a fully DNN-based implementation of CMT-SemCom proves ineffective due to the challenges of instability.

To overcome this, we introduce a hybrid-learning strategy that combines deep neural networks with kernelized-parametric machine learning techniques. This allows us to effectively implement IoPm while preserving the benefits of our cooperative multi-task semantic communication framework.

In summary, key contributions are:

- Extending the CMT-SemCom system to operate under rate-limited wireless channels, reflecting practical communication constraints.
- Proposing a separation-based design of the CU and SUs to achieve a more structured and effective formulation for constrained optimization.
- Addressing the limitations of fixed-prior regularization by adopting IoPm for more flexible and accurate KL divergence approximation.
- Introducing a hybrid-learning approach that integrates DNNs with parametric ML to robustly implement IoPm within the CMT-SemCom framework.

C. ORGANIZATION AND NOTATIONS

The rest of the paper is organized as follows. Section II presents probabilistic modeling of the proposed system model, followed by presenting two distinct objective functions that enable the separation-based design of the CU and SUs in Section II-B and II-C, respectively. Next, Section II-D describes the IoP method for enhanced approximation of the constrained problem in the SUs objective function and the proposed hybrid-learning approach. Section III presents simulation results evaluating the performance of the proposed CMT-SemCom across various datasets. Finally, Section IV concludes the paper highlighting the key findings. We also note that the notations used throughout this paper are listed in Table 1.

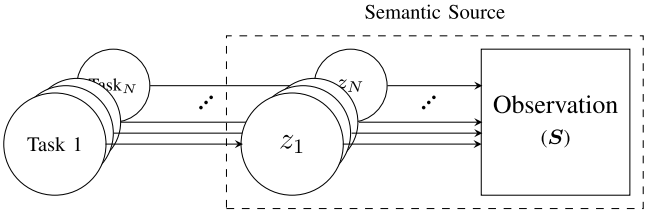


FIGURE 2. Probabilistic graphical modeling of the semantic source.

II. SYSTEM MODEL

This section explores the separation-based design for the proposed CMT-SemCom system model under constrained wireless channels. We begin by presenting the probabilistic modeling of the proposed framework in Section II-A. Following this, we formulate two distinct optimization problems: one focusing on the design of the CU, responsible for promoting cooperation amongst tasks, and the other targeting the design of the SUs, which is responsible for the joint semantic and channel coding (JSCC). We adopt the information maximization (Infomax) principle in Section II-B, while employing the information bottleneck (IB) approach in Section II-C to formulate the objective function for our constrained optimization problem. Next, Section II-D presents the IoPm and our hybrid-learning approach.

A. SYSTEM PROBABILISTIC MODELING

We begin by presenting our interpretation of the *semantic source* concept as discussed in [1]. We assume the existence of N independent tasks. Each task is entailed with its specific *semantic variable*, thus we have N semantic variables indicated by $\boldsymbol{z} = [\, z_1\, z_2\, \ldots\, z_N]$. We assume that our semantic variables are entailed with an observation, $\boldsymbol{S}$. We define the tuple of $(\boldsymbol{z}, \boldsymbol{S})$ as our semantic source, fully described by the probability distribution of $p(\boldsymbol{z}, \boldsymbol{S})$. Fig. 2 illustrates our interpretation using probabilistic graphical modeling [37] and a stack view for a better illustration. Such a definition enables the simultaneous extraction of multiple semantic variables based on a single observation. For instance, consider an image featuring both a tree and a number. One task may entail determining the presence of a tree, resulting in a binary semantic variable. Meanwhile, another task could focus on identifying the number within the image, yielding a multinomial semantic variable.

In this paper, we assume N tasks specify semantic variables to be delivered to their respective recipients through semantic decoders leveraging task-relevant information extracted by CU and SUs. It was demonstrated that when semantic variables share statistical relationships, CMT-SemCom enables cooperative processing and significantly improves performance in multi-task cases by utilizing common information [1].

Our system model has, on the transmitter side, the encoder split into one CU and multiple SUs. The CU encoder outputs a representation $\boldsymbol{c}$, which is the common relevant information extracted from the semantic source, via $p^{\,\mathrm{CU}}(\boldsymbol{c}|\boldsymbol{S})$. Next,

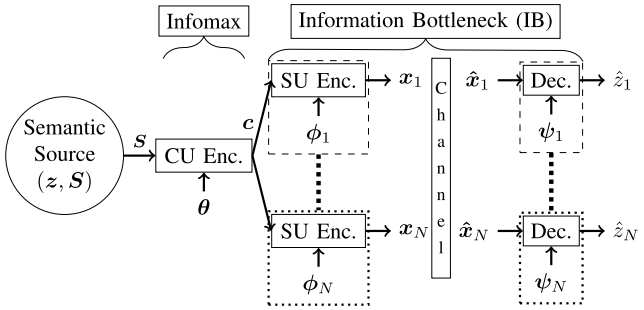


FIGURE 3. Illustration of the proposed separation-based design for the CMT-SemCom framework under rate-limit wireless channels.

each SU_n encodes $\boldsymbol{c}$ into a task-specific information $\boldsymbol{x}_n$ using $p^{\,\mathrm{SU}_n}(\boldsymbol{x}_n|\boldsymbol{c})$. These channel inputs are then transmitted through a rate-limited additive white Gaussian noise (AWGN) channel, resulting in received signals $\hat{\boldsymbol{x}}_n = \boldsymbol{x}_n + \boldsymbol{n}$, where $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}_d, \sigma_n^2 \boldsymbol{I}_d)$, and d indicates the number of channel uses (the limiting constraint) or in other words, the size of the encoded task-specific information, $\boldsymbol{x}_n \in \mathbb{R}^{d_n \times 1}$. On the Rx side, the semantic decoder $p^{\,\mathrm{Dec}_n}(\hat{z}_n|\hat{\boldsymbol{x}}_n)$ delivers the semantic variable z_i from $\hat{\boldsymbol{x}}_n$. The system model is also illustrated in Fig. 3.

Subsequently, the Markov representation of our system model for the n-th semantic variable is outlined as follows:

$$p\big(\hat{z}_n, \hat{\boldsymbol{x}}_n, \boldsymbol{x}_n, \boldsymbol{c}|\boldsymbol{S}\big) = p^{\,\mathrm{Dec}_n}\big(\hat{z}_n|\hat{\boldsymbol{x}}_n\big)\, p^{\,\mathrm{Channel}}\big(\hat{\boldsymbol{x}}_n|\boldsymbol{x}_n\big)\, p^{\,\mathrm{SU}_n}(\boldsymbol{x}_n|\boldsymbol{c}) p^{\,\mathrm{CU}}(\boldsymbol{c}|\boldsymbol{S}). \quad (1)$$

B. CU OBJECTIVE FUNCTION

To begin the separation-based design for the CU, which is shared amongst all SUs, we formulate the following optimization problem, adopting the Infomax principle.

$$p^{\,\mathrm{CU}}(\boldsymbol{c}|\boldsymbol{S})^{\star} = \arg \max_{p^{\,\mathrm{CU}}(\boldsymbol{c}|\boldsymbol{S})} I(\boldsymbol{c}; \boldsymbol{z}). \quad (2)$$

Hence, our objective is to maximize the mutual information between the CU output,$\boldsymbol{c}$, and the underlying semantic variables, $\boldsymbol{z} = [\, z_1\, z_2\, \ldots\, z_N]^T$, associated with the observation. Considering the availability of a sample set instead of the true distribution for $p(\boldsymbol{S}, \boldsymbol{z})$, we approximate the semantic source distribution with the corresponding available sample set [38]. Moreover, we employ the variational method, which is a way to approximate intractable computations based on some adjustable parameters, like weights in neural networks (NNs) [36]. The technique is widely used in machine learning, e.g., [39], and also in task-oriented communications, e.g., [24] and [25]. Thus, we approximate the posterior distribution $p^{\,\mathrm{CU}}(\boldsymbol{c}|\boldsymbol{S})$ by variational approximation using NN parameterized by $\boldsymbol{\theta}$. This approximation yields $p_{\boldsymbol{\theta}}^{\,\mathrm{CU}}(\boldsymbol{c}|\boldsymbol{S})$ and we present the CU objective function as follows.

$$\begin{aligned} \mathcal{L}^{\,\mathrm{CU}}(\boldsymbol{\theta}) &\approx I(\boldsymbol{c}; \boldsymbol{z}) \\ &\approx \int p(\boldsymbol{S}, \boldsymbol{z})\, p_{\boldsymbol{\theta}}^{\,\mathrm{CU}}(\boldsymbol{c}|\boldsymbol{S}) \log p(\boldsymbol{z}|\boldsymbol{c})\, d\boldsymbol{S}\, d\boldsymbol{z}\, d\boldsymbol{c} \\ &\approx \mathbb{E}_{p_{\boldsymbol{\theta}}^{\,\mathrm{CU}}(\boldsymbol{c}|\boldsymbol{S})}\big[\, \mathbb{E}_{p(\boldsymbol{S},\boldsymbol{z})}\big[\, \log p(\boldsymbol{z}|\boldsymbol{c})\, \big]\, \big]. \end{aligned} \quad (3)$$

Algorithm 1 Training the CU Encoder

Require: Preprocessed training dataset: $\{\boldsymbol{S}^{(m)}, z_1^{(m)}, \ldots, z_N^{(m)}\}_{m\in M}$, number of iterations T, batch sizes M_n

Ensure: The trained parameters $\boldsymbol{\theta}$ and $\boldsymbol{\Xi}$

for epoch $t = 1$ to T **do**
  for $n = 1$ to N **do**
    Randomly select a minibatch $\{(\boldsymbol{S}^{(m)}, z_n^{(m)})\}_{m=1}^{M_n}$
    Compute the mean vector $\{\boldsymbol{\mu}_{\boldsymbol{c}|\boldsymbol{S}^m}\}_{m=1}^{M_n}$
    Compute the standard deviation vector $\{\boldsymbol{\sigma}_{\boldsymbol{c}|\boldsymbol{S}^m}\}_{m=1}^{M_n}$
    for $m = 1$ to M_n **do**
      Sample the $\{\boldsymbol{\epsilon}^l\}_{l=1}^{L} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
      Compute $\boldsymbol{c}^{(m,l)} = \boldsymbol{\mu}_{\boldsymbol{c}|\boldsymbol{S}^m} + \boldsymbol{\sigma}_{\boldsymbol{c}|\boldsymbol{S}^m} \odot \boldsymbol{\epsilon}^l$
    end for
    Compute the log-likelihood $\log q_{\boldsymbol{\xi}_n}(z_n|\mathbf{c}^{(m,l)})$
  end for
  Compute the loss $\mathcal{L}^{\text{CU}}$ based on (6).
  Update parameters $\boldsymbol{\theta}$ and $\boldsymbol{\Xi}$ through backpropagation.
end for= 0

The outer expectation shows how the CU integrates the common knowledge extraction amongst the SUs and emphasizes our distinct approach caused by our architecture in cooperative processing. In addition, a detailed derivation for the infomax objective function of the CU can be found in [1]. Given the availability of the semantic source, denoted by the joint probability distribution $p(\boldsymbol{S}, z)$ and the posterior distribution $p_{\boldsymbol{\theta}}^{\text{CU}}(\boldsymbol{c}|\boldsymbol{S})$, the semantic space posterior $p(z|\boldsymbol{c})$ could be fully determined:

$$p(z|\boldsymbol{c}) = \int \frac{p(\boldsymbol{S}, z)\, p_{\boldsymbol{\theta}}^{\text{CU}}(\boldsymbol{c}|\boldsymbol{S})}{p_{\boldsymbol{\theta}}(\boldsymbol{c})}\, d\boldsymbol{S}. \tag{4}$$

Considering that $p_{\boldsymbol{\theta}}(\boldsymbol{c}) = \int p_{\boldsymbol{\theta}}^{\text{CU}}(\boldsymbol{c}|\boldsymbol{S})\, p(\boldsymbol{S})\, d\boldsymbol{S}$ could be also available. However, due to intractability of high dimensional integrals, we apply another variational approximation, replacing the true posterior distribution with its approximation $p_{\boldsymbol{\Xi}}(\boldsymbol{z}|\boldsymbol{c})$ where $\boldsymbol{\Xi} = [\,\boldsymbol{\xi}_1\, \boldsymbol{\xi}_2\, \ldots\, \boldsymbol{\xi}_N\,]^T$ is the parameters of the corresponding NNs of the auxiliary decoders for training the CU. Thus, the objective function in (3), is expressed as below.

$$\mathcal{L}^{\text{CU}}(\boldsymbol{\theta}, \boldsymbol{\Xi}) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}^{\text{CU}}(\boldsymbol{c}|\boldsymbol{S})}\big[\mathbb{E}_{p(\boldsymbol{S},z)}\big[\,\log p_{\boldsymbol{\Xi}}(z|\boldsymbol{c})\big]\,\big]. \tag{5}$$

Further, we approximate the expectations with Monte Carlo sampling following data-driven approach, given that there exists a dataset $\{\mathbf{S}^{(m)}, z_1^{(m)}, \ldots, z_N^{(m)}\}_{m=1}^{M}$ where M represents the dataset size and N denotes the number of available tasks. Thus, the empirical estimation of the objective function can be expressed as:

$$\mathcal{L}^{\text{CU}}(\boldsymbol{\theta}, \boldsymbol{\Xi}) \approx \frac{1}{L}\sum_{l=1}^{L}\left[\sum_{n=1}^{N}\left\{\frac{1}{M_n}\sum_{m=1}^{M_n}\log p_{\boldsymbol{\xi}_n}\left(z_n|\boldsymbol{c}^{(m,l)}\right)\right\}\right]. \tag{6}$$

Additionally, we have applied the reparameterization trick [36], to overcome the differentiability issues in the backpropagation of the objective function by introducing $\boldsymbol{c}^{(m,l)} = \boldsymbol{\mu}_{\boldsymbol{c}|\boldsymbol{S}^m} + \boldsymbol{\sigma}_{\boldsymbol{c}|\boldsymbol{S}^m} \odot \boldsymbol{\epsilon}^l$ where the auxiliary variable $\boldsymbol{\epsilon}^l \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ represents the element-wise product. Details on how the CU loss function is differentiable with respect to (w.r.t) $\boldsymbol{\theta}$ and the reparameterization trick are deferred to the Appendix A. In (6), L is the sample size of the reparameterization trick, fixed to one, signifying that the CU updates once, encompassing N specific features. M_n appears for the minibatch size of $\{(\mathbf{S}^{(m)}, z_n^{(m)})\}_{m=1}^{M_n}$ and for simplicity we assume that the minibatch sizes are equal across semantic variables, $M_n = M$. The training procedure for the CU is described in Algorithm 1.

C. SU OBJECTIVE FUNCTION

SUs are responsible for the JSCC, transmitting task-specific information such that the respective recipients can decode the intended semantic variables. To design the SU concerning the rate-limited wireless communication channel, we formulate the following constraint optimization problem.

$$\begin{aligned} p^{\,\text{SU}_n}(\boldsymbol{x}_n|\boldsymbol{c})^{\star} = \arg \max_{p^{\,\text{SU}_n}(\boldsymbol{x}_n|\boldsymbol{c})} \quad & I\big(\hat{\boldsymbol{x}}_n; z_n\big) \\ \text{subject to} \quad & I(\boldsymbol{x}_n; \boldsymbol{c}) \le R_n. \end{aligned} \tag{7}$$

Our formulation in (7), aims at maximizing the mutual information between the channel output, $\hat{\boldsymbol{x}}$, and the intended semantic variable while bounding the mutual information between the encoded signal, $\boldsymbol{x}$, and the input of the SUs, $\boldsymbol{c}$. The limited rate of the corresponding channel, R_n, is considered to limit the number of channel uses, d_n, by the n-th SU. This is how we adopt the information bottleneck method (IBM) [40], seeking the right balance between the inference accuracy and communication overhead using the mutual information as both an objective function and a constraint.

By employing the Lagrangian method [41] to optimization problem (7), we reformulate the objective function as:

$$\mathcal{L}^{\,\text{SU}_n} = I\big(\hat{\boldsymbol{x}}_n; z_n\big) - \lambda\,(I(\boldsymbol{x}_n; \boldsymbol{c}) - R_n). \tag{8}$$

We drop the constant term, λR_n, in (8) to get the simplified equivalent objective function. Moreover, as in Section II-B, the mutual information terms in (8) are generally intractable due to high-dimensional integrals. Also, following a data-driven approach, we leverage the variational approximation to form a tractable lower bound. Thus, expanding the mutual information terms and approximating the posterior distribution of the SU, $p^{\,\text{SU}_n}(\boldsymbol{x}_n|\boldsymbol{c})$, with NN parameterized by $\boldsymbol{\phi}_n$, yielding $p_{\boldsymbol{\phi}_n}^{\,\text{SU}_n}(\boldsymbol{x}_n|\boldsymbol{c})$, we end up with the following objective function:

$$\begin{aligned} \mathcal{L}^{\,\text{SU}_n}(\boldsymbol{\phi}_n) &= I\big(\hat{\boldsymbol{x}}_n; z_n\big) - \lambda\, I(\boldsymbol{x}_n; \boldsymbol{c}) \\ &\approx \mathbb{E}_{p_{\boldsymbol{\theta}}(z_n,\boldsymbol{c})}\Big[\mathbb{E}_{p_{\boldsymbol{\phi}_n}^{\,\text{SU}}(\boldsymbol{x}_n|\boldsymbol{c})}\Big[\mathbb{E}_{p(\hat{\boldsymbol{x}}_n|\boldsymbol{x}_n)}\big[\,\log p\big(z_n|\hat{\boldsymbol{x}}_n\big)\big]\,\Big] \\ &\qquad - \lambda\, KL\Big(p_{\boldsymbol{\phi}_n}^{\,\text{SU}}(\boldsymbol{x}_n|\boldsymbol{c})\,\|\,p(\boldsymbol{x}_n)\Big)\Big], \end{aligned} \tag{9}$$

where in (9), the outer expectation represents the effect of the pre-trained CU. The detailed derivation of the objective function above is deferred to Appendix B. Owing to the pre-trained CU, $p_{\boldsymbol{\theta}}(z_n, \boldsymbol{c})$ is already available as:

$$p_{\boldsymbol{\theta}}(z_n, \boldsymbol{c}) = \int p(z_n, \boldsymbol{S})\, p_{\boldsymbol{\theta}}(\boldsymbol{c}|\boldsymbol{S})\, d\boldsymbol{S}. \tag{10}$$

The log-likelihood function appearing in the first term of the objective function (9) can be fully described when $p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})$ is available by:

$$p(z_n|\hat{\boldsymbol{x}}_n) = \int \frac{p_{\boldsymbol{\theta}}(z_n, \boldsymbol{c})\, p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})\, p(\hat{\boldsymbol{x}}_n|\boldsymbol{x}_n)}{p(\hat{\boldsymbol{x}}_n)}\, d\boldsymbol{c}\, d\boldsymbol{x}, \tag{11}$$

however, same as (4), we must once more use approximations and replace the true likelihood distribution with its approximated version $p_{\boldsymbol{\psi}_n}(z_n|\hat{\boldsymbol{x}}_n)$, where $\boldsymbol{\psi}_n$ is the parameters of the corresponding NN for the n-th task-specific decoder. Therefore, the objective function is expressed as:

$$\begin{aligned}\mathcal{L}^{\mathrm{SU}_n}&\left(\boldsymbol{\phi}_n, \boldsymbol{\psi}_n\right)\\ &\approx \mathbb{E}_{p_{\boldsymbol{\theta}}(z_n,\boldsymbol{c})}\Big[\, \mathbb{E}_{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\hat{\boldsymbol{x}}_n|\boldsymbol{c})}\big[\log q_{\boldsymbol{\psi}_n}(z_n|\hat{\boldsymbol{x}}_n)\big]\\ &\qquad - \lambda\, KL\Big(p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})\|p(\boldsymbol{x}_n)\Big)\Big].\end{aligned} \tag{12}$$

As we have a DNN-based implementation followed by an E2E learning fashion, improved to be effective for task-oriented communication [42], we emphasize performing JSCC by the SU encoders by $p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\hat{\boldsymbol{x}}_n|\boldsymbol{c}) = \int p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})\, p(\hat{\boldsymbol{x}}_n|\boldsymbol{x}_n)\, d\boldsymbol{x}_n$. This means we are taking semantic and channel statistics into account in a joint manner. In addition, we note that a detailed discussion on the approximation error analysis for the objective function in (12) is provided in Appendix C.

For the regularization term in (12), where the KL divergence appears, adopting a variational marginal posterior distribution for $p(\boldsymbol{x}_n)$, which can be also called the *prior distribution* of the SU's output space, is necessary. Fixing the marginal posterior distribution, or the prior, to choices such as a standard Gaussian distribution, introduced and mostly used in training variational autoencoder structures [36], or a log-uniform distribution, which is favored for its sparsity-inducing properties, has been the most common approach taken in the literature, e.g., [25], [43], [44] for adopting standard Gaussian prior and [24], [45] for log-uniform. The primary motivation behind these choices is that they allow the KL divergence term to be computed in closed form, greatly simplifying optimization. We refer to this approach as the explicit prior (EP) method, since the prior is explicitly fixed to either a Gaussian or log-uniform distribution. However, this convenience comes at the cost of sub-optimality as the prior is restricted for mathematical tractability rather than for faithfully modeling the data. To address this limitation, we modify the loss function to incorporate density ratio estimation, enabling a more flexible and accurate approximation of the KL divergence and, in turn, a better approximation of the objective function.

D. IMPLICIT OPTIMAL PRIOR METHOD

To optimally deal with the regularization term, we modify the KL divergence in (12) as follows:

$$\begin{aligned}&KL\Big(p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})\|p(\boldsymbol{x}_n)\Big)\\ &= \mathbb{E}_{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})}\left[\log \frac{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})}{p(\boldsymbol{x}_n)} \cdot \frac{q(\boldsymbol{x}_n)}{q(\boldsymbol{x}_n)}\right]\\ &= KL\Big(p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})\|q(\boldsymbol{x}_n)\Big) - \mathbb{E}_{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})}\left[\log \frac{p(\boldsymbol{x}_n)}{q(\boldsymbol{x}_n)}\right],\end{aligned} \tag{13}$$

where $q(\boldsymbol{x}_n)$ is an arbitrary distribution, typically chosen as either a standard Gaussian or a log-uniform distribution, to ensure that the KL divergence can be computed in closed form. This trick, introduced in [46] for a variational auto-encoder, enables us to implicitly manage the prior distribution by estimating the density ratio $p(\boldsymbol{x}_n)/q(\boldsymbol{x}_n)$ and prevent fixing a distribution. Reference [46] estimates the prior using a DNN-based classifier accompanied by several regularization techniques to fine-tune the estimator. In our investigations, we faced many issues while adopting the method in [46] to our multi-tasking SemCom framework. The issues include the convergence of the DNN-based classifier and the complexity of fine-tuning due to the existence of several regularization parameters.

To overcome these issues, we propose using classical parametric ML methods to estimate density ratios by introducing a *hybrid-learning approach*. To develop our density ratio estimation for the implicit optimal prior method (IoPm) in CMT-SemCom, we follow the *probabilistic classification* approach amongst other approaches of density ratio estimation [47]. Using the probabilistic classification approach has advantages such as straightforward implementation and the possibility of direct use of a standard classification algorithm.

Specifically, we train a probabilistic binary classifier to distinguish between samples drawn from the arbitrary distribution, $q(\boldsymbol{x}_n)$, and samples drawn from the distribution produced by the semantic encoder, $p(\boldsymbol{x}_n)$. The key insight is that the classifier's outputs can be transformed to approximate the density ratio, which in turn allows us to compute the regularization term without requiring an explicit prior.

For this, we first sample from $q(\boldsymbol{x}_n)$ and assign labels $y = 0$ to them. Next, labels $y = 1$ go to samples from $p(\boldsymbol{x}_n)$ which are available using ancestral sampling [38] from the output of our encoder, $p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\mathbf{x}_n|\mathbf{c})$. Then, $p(\boldsymbol{x}_n|y)$ is defined as:

$$p(\boldsymbol{x}_n|y) = \begin{cases} p(\boldsymbol{x}_n) & y = 1,\\ q(\boldsymbol{x}_n) & y = 0.\end{cases} \tag{14}$$

Thus, our density ratio can be expressed as:

$$\begin{aligned}r(\boldsymbol{x}_n) &= \frac{p(\boldsymbol{x}_n)}{q(\boldsymbol{x}_n)} = \frac{p(\boldsymbol{x}_n|y=1)}{p(\boldsymbol{x}_n|y=0)}\\ &= \frac{p(y=1|\boldsymbol{x}_n)\, p(y=0)}{p(y=0|\boldsymbol{x}_n)\, p(y=1)} = \frac{p(y=1|\boldsymbol{x}_n)}{p(y=0|\boldsymbol{x}_n)},\end{aligned} \tag{15}$$

Algorithm 2 Density Ratio Estimation Using Kernelized LR

Require: Samples $\boldsymbol{x}_p \sim p(\mathbf{x}_n)$, samples $\boldsymbol{x}_q \sim q(\mathbf{x}_n)$, kernel bandwidth σ_k

Ensure: Estimated density ratio $\hat{r}(\boldsymbol{x}_n)$

Step 1: Generate labels for samples:

$\mathbf{y}_p = \mathbf{1}_n$, $\mathbf{y}_q = -\mathbf{1}_n$

Step 2: Combine samples and labels:

$\boldsymbol{X} = [\boldsymbol{x}_p; \boldsymbol{x}_q]$

$\mathbf{y} = [\mathbf{y}_p; \mathbf{y}_q]$

Step 3: Compute the Gaussian kernel:

$K = \exp\left(-\frac{\|\boldsymbol{X}-\boldsymbol{X}'\|^2}{2\sigma^2}\right)$

Step 4: Train logistic regression model:

Fit logistic regression on K with labels $\mathbf{y}$, and get $\hat{\boldsymbol{\omega}}$

Step 5: Estimate density ratio for new data points $\boldsymbol{X}$:

Compute the kernel matrix K_{new} between saved $\boldsymbol{X}'$ from training and the new data $\boldsymbol{X}$

$\hat{r}(\boldsymbol{x_n}) = \exp\left(K_{\text{new}} \cdot \hat{\boldsymbol{\omega}}\right)$

Step 6: Return estimated density ratio $\hat{r}(\boldsymbol{x})$

where in (15), we cancel $p(y=0)$ with $p(y=1)$ since we draw an equal number of samples from both distributions. Therefore, given an estimator of the posterior probability $\hat{p}(y|\boldsymbol{x}_n)$, we can estimate the density ratio. In this work, we leverage *logistic regression* (LR) classification that employs a parametric model of the following for the posterior distribution:

$$p(y|\boldsymbol{x}_n;\boldsymbol{\omega}) = \left(1+\exp\left(-y\Omega(\boldsymbol{x}_n)^T\boldsymbol{\omega}\right)\right)^{-1}, \tag{16}$$

where $\Omega(\boldsymbol{x}_n)$ is a basis function and $\boldsymbol{\omega}$ is the parameter vector. Our LR model parameter is learned so that the penalized log-likelihood is maximized:

$$\hat{\boldsymbol{\omega}} = \arg\max_{\boldsymbol{\omega}}\left[\sum_{k=1}^{K}\log\left(1+\exp\left(-y_k\Omega\left(\boldsymbol{x}_n^{(k)}\right)^T\boldsymbol{\omega}\right)\right)+\gamma\boldsymbol{\omega}^T\boldsymbol{\omega}\right], \tag{17}$$

In (17), the term $\gamma\boldsymbol{\omega}^T\boldsymbol{\omega}$ serves as a regularization term for the LR objective function, preventing overfitting. A key advantage of the LR objective function in (17) is its convexity, which guarantees that *gradient descent* (GD) methods can converge to the global optimum [48]. Finally, using (15) and (16), our density ratio estimator (DRE) is expressed as:

$$\hat{r}_{LR}(\boldsymbol{x}_n) = \frac{1+\exp\left(\Omega(\boldsymbol{x}_n)^T\hat{\boldsymbol{\omega}}\right)}{1+\exp\left(-\Omega(\boldsymbol{x}_n)^T\hat{\boldsymbol{\omega}}\right)} = \exp\left(\Omega(\boldsymbol{x}_n)^T\hat{\boldsymbol{\omega}}\right). \tag{18}$$

For the DRE in (18), we use the *Gaussian kernel* for the basis function with kernel bandwidth, σ_k as:

$$K(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\sigma_k^2}\right), \tag{19}$$

where $\boldsymbol{x}'$ denotes the stored samples of $\boldsymbol{x}_n$ from both distributions, obtained in the previous step, that we store for the next inference step of the DRE. This reflects our use of a *memory-based* method [38] for the DRE that involves storing the samples used in training to make inferences for future data. A detailed description of the DRE procedure is provided in Algorithm 2, with further discussions in Section III.

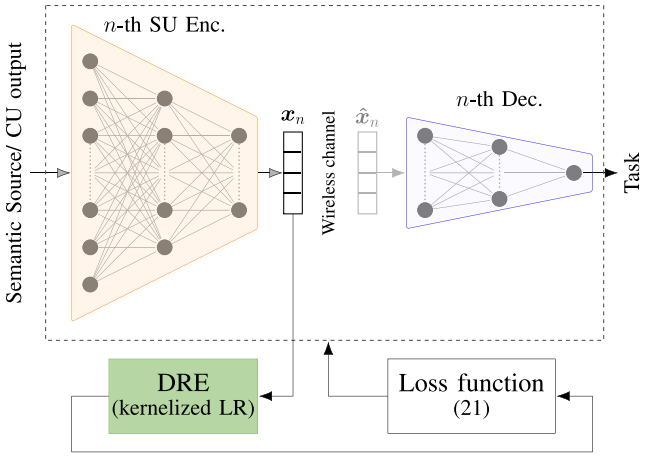


FIGURE 4. Illustration of the proposed hybrid-learning approach for the *n*-th SU.

By employing the kernelized LR, we implement the IoPm within the CMT-SemCom framework to more effectively handle the regularization term, representing the communication overhead introduced by the rate-limited wireless channels. Fig. 4, illustrates our hybrid-learning approach, which combines this classical kernelized DRE with our DNN-based semantic transmission. The output of the n-th SU, $\boldsymbol{x}_n$, is sampled and provided to the DRE. In the DRE unit, the samples from $q(\boldsymbol{x}_n)$ are also drawn, and the regularization is estimated by (18). This estimate is then used to update the SU's objective function in each training iteration. As training progresses, improved encoder outputs lead to more accurate density ratio estimates, which in turn refine the overall loss function.

Applying the discussed IoPm, the approximated objective function in (12) becomes:

$$\begin{aligned}\mathcal{L}^{\text{SU}_n}&(\boldsymbol{\phi}_n,\boldsymbol{\psi}_n)\\ &\approx \mathbb{E}_{p_{\boldsymbol{\theta}}(z_n,\boldsymbol{c})}\Big[\mathbb{E}_{p_{\boldsymbol{\phi}_n}^{\text{SU}}(\hat{\boldsymbol{x}}_n|\boldsymbol{c})}\left[\log q_{\boldsymbol{\psi}_n}\left(z_n|\hat{\boldsymbol{x}}_n\right)\right]\\ &\quad-\lambda\left\{KL\left(p_{\boldsymbol{\phi}_n}^{\text{SU}}(\boldsymbol{x}_n|\boldsymbol{c})\|q(\boldsymbol{x}_n)\right)-\mathbb{E}_{p_{\boldsymbol{\phi}_n}^{\text{SU}}(\boldsymbol{x}_n|\boldsymbol{c})}\left[\log\hat{r}(\boldsymbol{x}_n)\right]\right\}\Big].\end{aligned} \tag{20}$$

Given that a minibatch of $\{z_n^{(m)},\boldsymbol{c}^{(m)}\}_{m=1}^{M_n}$ can be selected from the joint distribution $p_{\boldsymbol{\theta}}(z_n,\boldsymbol{c})$ and leveraging the Monte Carlo sampling as we did for (6), we end up with the empirical estimation of the objective function:

$$\begin{aligned}\mathcal{L}^{\text{SU}_n}(\boldsymbol{\phi}_n,\boldsymbol{\psi}_n) \approx \frac{1}{M_n}\sum_{m=1}^{M_n}\Bigg\{&\frac{1}{L}\sum_{l=1}^{L}\left[\log q_{\boldsymbol{\psi}_n}\left(z_n|\hat{\boldsymbol{x}}_n^{(m,l)}\right)\right]\\ &-\lambda\left[KL\left(p_{\boldsymbol{\phi}_n}^{\text{SU}}(\boldsymbol{x}_n^{(m)}|\boldsymbol{c})\|q(\boldsymbol{x}_n)\right)-\frac{1}{L}\sum_{l=1}^{L}\hat{r}\left(\boldsymbol{x}_n^{(m,l)}\right)\right]\Bigg\}.\end{aligned} \tag{21}$$

It is worth mentioning that in (21), we apply the reparameterization trick to overcome the differentiability issues in the backpropagation as previously discussed in Appendix A. For the term, representing the corresponding channel rate, the reparameterization trick exists as $\boldsymbol{x}_n^{(m,l)} = \boldsymbol{\mu}_{\boldsymbol{x}_n|\boldsymbol{c}^m} + \boldsymbol{\sigma}_{\boldsymbol{x}_n|\boldsymbol{c}^m} \odot \boldsymbol{\epsilon}^l$.

It is important to note that the integration of parametric DRE with DNN-based SUs is feasible in practice, particularly when using *stochastic gradient descent* (SGD) as the optimizer for training the SU networks with the objective function in (20). The use of SGD enables a key simplification by allowing us to treat the DRE as a fixed, non-trainable component during the SU training. Specifically, the term, $\mathbb{E}_{p_{\boldsymbol{\phi}_n}^{\text{SU}}(\boldsymbol{x}_n|\boldsymbol{c})}[\log \hat{r}(\boldsymbol{x}_n)]$, is not included in the backpropagation process for updating the SU parameters $\boldsymbol{\phi}_n$. This decoupling is what enables our hybrid-learning approach, where the DRE is trained separately using classical methods, while the DNN-based SUs are trained via SGD. Without this separation, the DRE would need to be differentiable and involved in gradient updates, significantly complicating the training pipeline. A detailed explanation of why this integration is compatible with SGD-based optimization is provided in Appendix D.

1) KL DIVERGENCE CLOSED-FORM EXPRESSION

Finally, the last step is to manage the KL divergence term in eq. (21). We assumed the Gaussian distribution for our SU encoder such that $p_{\boldsymbol{\phi}_n}^{\text{SU}}(\boldsymbol{x}_n|\boldsymbol{c}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}_n|\boldsymbol{c}^m}, \boldsymbol{\sigma}_{\boldsymbol{x}_n|\boldsymbol{c}^m}\mathbf{I})$. Thus, the proper selection of the arbitrary prior $q(\boldsymbol{x}_n)$, e.g., standard Gaussian or log-uniform, can result in a closed-form solution for the KL term in eq. (21). In addition, since we compare our proposed method with the two most widely adopted fixed priors: the standard Gaussian and the log-uniform distribution in the EP method in Section III, we need to directly calculate the KL term in eq. (12) for fixed $p(\boldsymbol{x}_n)$. Here, we present the KL divergence calculation for both cases.

Standard Gaussian Prior: When the arbitrary distribution $q(\boldsymbol{x}_n)$ in eq. (21) (or, equivalently, the fixed prior in the EP method $p(\boldsymbol{x}_n)$ in eq. (12)) is chosen as the standard Gaussian, the KL divergence reduces to the well-known closed-form expression between two Gaussian distributions:

$$KL\left(p_{\boldsymbol{\phi}_n}^{\text{SU}}(\boldsymbol{x}_n|\boldsymbol{c}) \| q(\boldsymbol{x}_n)\right) = \frac{1}{2}\left(\log \frac{1}{\sigma_{x_n|\boldsymbol{c}}^2} + \sigma_{x_n|\boldsymbol{c}}^2 + \mu_{x_n|\boldsymbol{c}}^2 - 1\right). \tag{22}$$

We note that this results from computing the KL divergence between the output distribution of each SU encoder, $p_{\boldsymbol{\phi}_n}^{\text{SU}}(\boldsymbol{x}_n|\boldsymbol{c})$ and its corresponding prior $q(\boldsymbol{x}_n)$ (or, equivalently, $p(\boldsymbol{x}_n)$ in the EP method). In other words, the SU objective $(\mathcal{L}^{\text{SU}_n}(\boldsymbol{\phi}_n, \boldsymbol{\psi}_n))$ is per SU and is calculated for the n-th SU.

Log-Uniform Prior: Alternately, when $q(\boldsymbol{x}_n)$ (or equivalently, $p(\boldsymbol{x}_n)$ in the EP method) is selected as a log-uniform distribution, the KL divergence can also be approximately expressed in closed form by taking advantage of the results of [49], [50] as follows:

$$\begin{aligned} &KL\left(p_{\boldsymbol{\phi}_n}^{\text{SU}}\left(x_n^{(i)}|\boldsymbol{c}\right) \| q\left(x_n^{(i)}\right)\right) \\ &= \tfrac{1}{2}\log \alpha_i - \mathbb{E}_{\epsilon\sim\mathcal{N}(1,\alpha_i)}\left[\log|\epsilon|\right] + C \\ &\approx k_1\,\sigma\left(k_2 + k_3 \log \alpha_i\right) - \tfrac{1}{2}\log\left(1 + \alpha_i^{-1}\right) + C, \end{aligned} \tag{23}$$

where

$$\alpha_i = \frac{\sigma_{x_n^{(i)}|\boldsymbol{c}}^2}{x_n^{(i)^2}}, \quad k_1 = 0.63576, \quad k_2 = 1.87320, \quad k_3 = 1.48695.$$

and C is a constant. Besides, $x_n^{(i)}$ is the i-th dimension in $\boldsymbol{x}_n$, and $\sigma(\cdot)$ denotes the sigmoid function.

Consequently, the empirical approximation of the objective function in (12) is calculated as above. The training procedure for the n-th SU adopting the standard Gaussian for the arbitrary prior is described in Algorithm 3.

Algorithm 3 Training the n-th SU Encoder and Decoder

Require: Preprocessed training dataset: $\{\boldsymbol{S}^{(m)}, z_1^{(m)}, \ldots, z_N^{(m)}\}_{m\in M}$, optimized parameters $\boldsymbol{\theta}$, number of iterations T, batch sizes M_n

Ensure: The trained parameters $\boldsymbol{\phi}_n$, $\boldsymbol{\psi}_n$, and $\boldsymbol{\omega}$

for epoch $t = 1$ to T **do**
  Randomly select a minibatch $\{(\boldsymbol{S}^{(m)}, z_n^{(m)})\}_{m=1}^{M_n}$
  Extract $\{\boldsymbol{c}^m\}_{m=1}^{M_n}$ from the learned $p_{\boldsymbol{\theta}}^{\text{CU}}(\boldsymbol{c}|\boldsymbol{S})$
  compute the mean vector $\{\boldsymbol{\mu}_{\boldsymbol{x}_n|\boldsymbol{c}^m}\}_{m=1}^{M_n}$ and the standard deviation vector $\{\boldsymbol{\sigma}_{\boldsymbol{c}|\boldsymbol{S}^m}\}_{m=1}^{M_n}$
  for $m = 1$ to M_n **do**
    Sample the $\{\boldsymbol{\epsilon}^{(m,l)}\}_{l=1}^{L} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    Compute $\boldsymbol{x}_n^{(m,l)} = \boldsymbol{\mu}_{\boldsymbol{x}_n|\boldsymbol{c}^m} + \boldsymbol{\sigma}_{\boldsymbol{x}_n|\boldsymbol{c}^m} \odot \boldsymbol{\epsilon}^{(m,l)}$
  end for
  Compute the log-likelihood $\log q_{\boldsymbol{\psi}_n}(z_n|\hat{\boldsymbol{x}}_n^{(m,l)})$
  Compute the density ratio based on Algorithm 2
  Compute the gradients of $\mathcal{L}^{\text{SU}_n}$ in (21)
  Update parameters $\boldsymbol{\phi}_n$ and $\boldsymbol{\psi}_n$
  Compute the total loss value $\mathcal{L}^{\text{SU}_n}$ based on (21).
end for=0

III. SIMULATION RESULTS

To evaluate the effectiveness of our proposed CMT-SemCom design over rate-limited wireless channels using the hybrid-learning framework, we consider two representative tasks in our multi-label tasks paradigm: binary and categorical classification. These correspond to two different semantic variables, modeled as $z_1 \sim$ *Bernoulli* and $z_2 \sim$ *Multinomial*. We begin by assessing the accuracy of our proposed density ratio estimator. Then, we present the overall performance of the CMT-SemCom across various datasets. In addition, we examine system behavior under different levels of channel constraint. Finally, we compare the proposed IoPm with two

TABLE 2. Encoder-decoder NN architecture for the proposed CMT-SemCom.

(a) NN structure for the MNIST dataset

	Layer	Properties
CU	Dense	size: 256, activation: ReLU
	Dense	size: 256, activation: ReLU
	Dense ($\boldsymbol{\mu_c}$)	size: 128, activation: Linear
	Dense ($\boldsymbol{\sigma_c}$)	size: 128, activation: Linear
Dec$_{\text{aux}}$	Dense	size: 128, activation: ReLU
	Dense (T_1)	size: 1, activation: Sigmoid
	Dense (T_2)	size: 10, activation: Softmax
SU	Dense	size: 64, activation: ReLU
	Dense	size: 64, activation: ReLU
	Dense ($\boldsymbol{\mu}_{\boldsymbol{x}_n}$)	size: 32, activation: Tanh
	Dense ($\boldsymbol{\sigma}_{\boldsymbol{x}_n}$)	size: 32, activation: Sigmoid
Dec	Dense	size: 32, activation: ReLU
	Dense (T_1)	size: 1, activation: Sigmoid
	Dense (T_2)	size: 10, activation: Softmax

(b) NN structure for the CIFAR-10 dataset

	Layer	Properties
CU	Conv2D	filter size: 32, kernel size: (8,8), activation: ReLU
	Conv2D	filter size: 32, kernel size: (8,8), activation: ReLU
	MaxPooling2D	pool size: (2,2)
	Dropout	dropout rate: 0.1
	Conv2D	filter size: 32, kernel size: (8,8), activation: ReLU
	MaxPooling2D	pool size: (2,2)
	Dropout	dropout rate: 0.2
	Conv2D	filter size: 32, kernel size: (8,8), activation: ReLU
	MaxPooling2D	pool size: (2,2)
	Dropout	dropout rate: 0.2
	Flatten	-
	Dense ($\boldsymbol{\mu_c}$)	size: 256, activation: Linear
	Dense ($\boldsymbol{\sigma_c}$)	size: 256, activation: Linear
Dec$_{\text{aux}}$	Dense	size: 256, activation: ReLU
	Dense	size: 128, activation: ReLU
	Dropout	dropout rate: 0.2
	Dense (T_1)	size: 1, activation: Sigmoid
	Dense (T_2)	size: 10, activation: Softmax
SU	Dense	size: 256, activation: ReLU
	Dense	size: 256, activation: ReLU
	Dense ($\boldsymbol{\mu}_{\boldsymbol{x}_n}$)	size: 128, activation: Tanh
	Dense ($\boldsymbol{\sigma}_{\boldsymbol{x}_n}$)	size: 128, activation: Sigmoid
Dec	Dense	size: 128, activation: ReLU
	Dense (T_1)	size: 1, activation: Sigmoid
	Dense (T_2)	size: 10, activation: Softmax

widely used baselines that follow explicit fixed prior (EP) method (standard Gaussian and log-uniform prior).[1]

[1]The simulation code of this paper is available at https://github.com/ant-uni-bremen/CMT-SemCom_IoPm.

TABLE 3. Specification of the parametric kernelized DRE.

	Component	Description
	Model	Logistic Regression
	Kernel	Radial Basis Function kernel
	Training	Quasi-Newton Method
MNIST	Kernel BW	$\sigma_k = 1.9$
	Regularization	$\gamma = 1.5$
	Sample size	2000
CIFAR-10	Kernel BW	$\sigma_k = 5.0$
	Regularization	$\gamma = 2.0$
	Sample size	4000

A. SIMULATION SETUP

We examine the proposed framework across two widely adopted datasets in semantic/task-oriented communication [24], [25], [45], [51], [52], [53], [54]. The MNIST dataset of handwritten digits [55], contains 60,000 images for the training set and 10,000 samples for the test set. Moreover, the CIFAR-10 dataset [56] consists of 60000, 32×32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. For a specific number of tasks denoted by T, we shape our semantic source as $\{\boldsymbol{S}^{(m)}, z_1^{(m)}, \ldots, z_T^{(m)}\}_{m=1}^{M}$. The implemented DNN structure and the specification of the parameterized DRE are listed in Table 2 and Table 3, respectively. The specifications are found heuristically such that the performance is maximized.

B. DRE PERFORMANCE

To implement the IoPm using the density ratio trick, we initially explored a DNN-based approach for the DRE, motivated by the generalization capabilities of DNNs. However, within the context of our CMT-SemCom framework, we encountered various challenges related to convergence and hyperparameter tuning. For simple cases like a single variational autoencoder, i.e., [46], many techniques such as dropout, dynamic binarization, early stopping, etc., are employed together to fine-tune the estimator. However, these strategies failed to stabilize training in our more complex multi-task setting.

As a result, we turned to a classical parametric ML method, which offers a simpler and more reliable implementation. This approach not only avoids the instability of DNN training but also provides the potential for optimal estimation under correct model specification, making it well-suited for our hybrid-learning framework.

We first evaluate the behavior of the DRE in a simple one-dimensional setting. As shown in Fig. 5, the estimator accurately captures the density ratio between two univariate Gaussian distributions, $x_1 \sim \mathcal{N}(0, 1)$ and $x_2 \sim \mathcal{N}(1, 2)$. The performance changes depending on the DRE's specification, i.e., sample size, kernel, etc., and the specification used in our evaluations is included in the figures.

To inspect how dimensionality affects the performance of the proposed DRE, we extend this analysis to multivariate

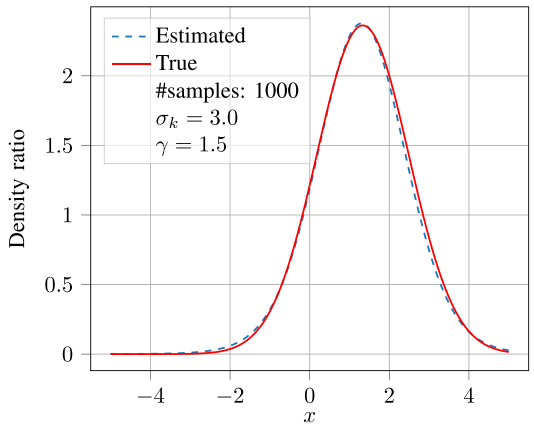
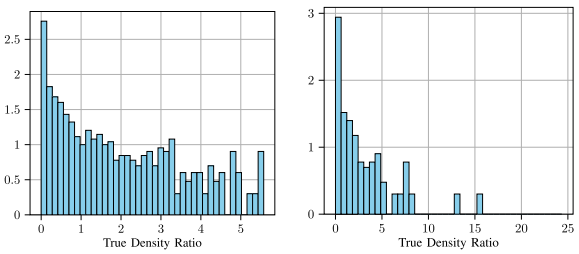


FIGURE 5. Performance of the proposed DRE for the scalar data.



(a) 2D (b) 4D

FIGURE 6. The histogram of the true density ratios for different data dimensions.

cases with increasing dimensions. Figs. 7 (a, b, c) present scatter plots of the estimated density ratios in 1D, 2D, and 4D. The evaluations in these figures, are for estimating the density ratio of two multivariate Gaussian distribution. For instance, for the 4D case the distributions are $\mathbf{x}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, where $\boldsymbol{\mu}_1 = \begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix}^\top$ and $\boldsymbol{\Sigma}_1 = \mathbf{I}_{d=4}$, and $\mathbf{x}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, where $\boldsymbol{\mu}_2 = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}^\top$ and $\boldsymbol{\Sigma}_2 = 4 \cdot \mathbf{I}_{d=4}$. We observe that as the dimension increases, the accuracy of the DRE decreases.

This decline in performance is further illustrated in Fig. 6, which shows the histograms of the true density ratios for 2D and 4D cases. We observe that as the dimension increases, the values of the density ratios tend to concentrate around lower values. This concentration makes it difficult for the estimator to distinguish between regions of high and low density, especially under limited sampling.

Further, we evaluate the impact of varying sample sizes on the DRE performance. To investigate the effect of sample size, we evaluate the DRE in the 1D case and keep all other specifications the same across different sample sizes. Figs. 7 (a, b, c) illustrate this impact for sample sizes of 500, 1000, and 2000. We observe that the DRE benefits from increased sample sizes up to a point but it degrades with more samples due to the fact that the kernel becomes noise sensitive, and can have memory expansion as well. For investigation on the impact of other DRE specifications, e.g., kernel bandwidth, we invite the interested readers to check our published simulation codes available at https://github.com/ant-uni-bremen/CMT-SemCom_IoPm.

We note that for our further experimental evaluations, we employed a grid search with cross-validation to select the best combination for the DRE specifications, and these specifications are listed in Table 2 for different datasets. Moreover, we examined other kernels for the basis function, such as the *Polynomial kernel* and the *Sigmoid kernel*, to inspect their ability to capture the complex interactions in high dimensions, but the Gaussian kernel still had the best performance.

C. IMPACT OF THE IoPM ON COOPERATIVE MULTI-TASKING

We evaluate the effectiveness of the proposed rate-limited CMT-SemCom enabled by IoPm by measuring task execution error rates. Specifically, we compare two scenarios:

- w.CU (with CU): Both SUs cooperate through the CU to execute their tasks.
- w.o.CU (without CU): CU is omitted and each SU execute its task independently, using the semantic source directly as input without any cooperation.

Fig. 8(a) presents the comparison for the MNIST dataset. The results clearly show that the cooperative processing case (w.CU), CMT-SemCom, improves performance for both Task1 and Task2. In contrast, w.o.CU exhibits a steadier and slower improvement. This behavior is explained by our hybrid-learning strategy. In the early stages of training, the DRE struggles to provide accurate estimates because the encoder in the SU has not yet converged. As training progresses and the SU starts producing more meaningful outputs, the DRE becomes more effective, leading to visible performance gains.

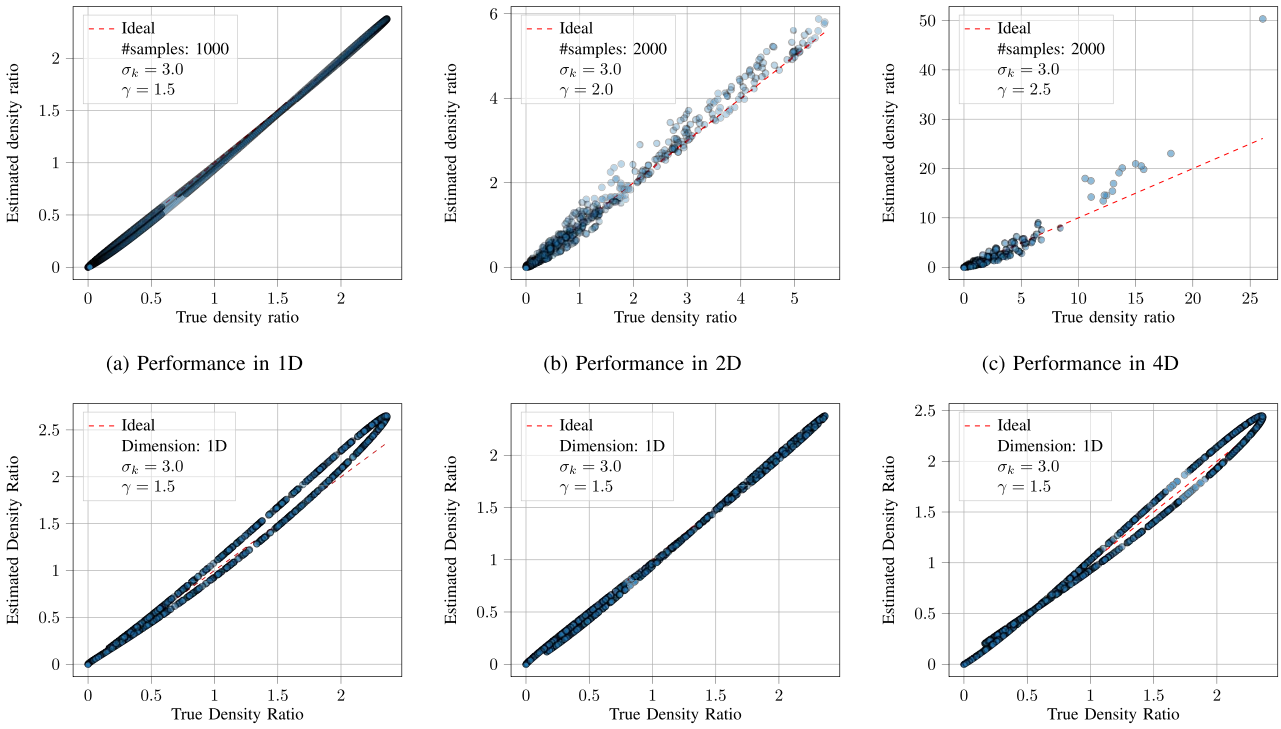
Moreover, the cooperative processing enabled by the CMT-SemCom framework accelerates this convergence by allowing tasks to share semantic information, thereby reducing the number of iterations required to achieve high accuracy.

A similar behavior is observed for the CIFAR dataset, as shown in Fig. 8(b). While cooperation still improves performance, the gap between w.CU and w.o.CU is smaller compared to the MNIST case. Nevertheless, the results confirm that even for complex datasets, IoPm-based CMT-SemCom facilitates the task execution performance.

D. IMPACT OF THE CHANNEL CONSTRAINT

We investigate the influence of rate-limited wireless channels on the performance of the proposed CMT-SemCom framework by varying the number of the available channel uses (d). We note that d is a translation of the rate constraint and a quantitative representation of the limit. The average task execution error rate for both tasks serves as the evaluation metric to capture how constrained bandwidth affects system accuracy.

Fig. 9(a) and Fig. 9(b) illustrate this effect for the MNIST and CIFAR datasets, respectively.



(a) Performance in 1D (b) Performance in 2D (c) Performance in 4D

(d) Performance with #samples=500 (e) Performance with #samples=1000 (f) Performance with #samples=2000

FIGURE 7. **Performance of the proposed DRE for different specifications of data dimensions and number of samples.**

For the MNIST dataset, a clear performance degradation is observed as the channel becomes more constrained. The task execution error increases from approximately 2% at 64 channel uses to over 36% at 4 channel uses. This behavior highlights the sensitivity of the system's performance to channel limitations.

For the CIFAR dataset, the behavior differs. While performance degrades at extreme channel limitations (e.g., 4 and 8 channel uses), the error rate remains relatively stable across a broad range. Moreover, the steeper drop in performance below the 16 channel uses for CIFAR dataset indicates a threshold effect. Once the encoded representation faces a specific limit, the loss of information becomes significant, and a sharp drop in task execution quality takes place. The impact is less dramatic for the MNIST.

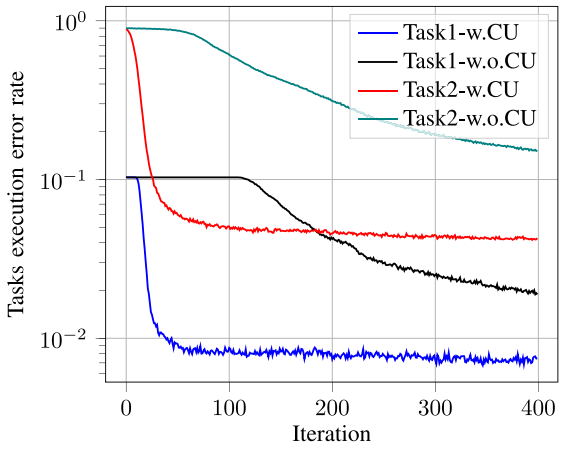
E. HYBRID-LEARNING IoPM VS. EP

Finally, we compare our proposed hybrid-learning based IoPm approach with the two widely used fixed priors in EP method, which are the standard Gaussian and the log-uniform prior. As shown in Figs. 10(a) and 10(b), our hybrid-learning-based IoPm consistently outperforms the EP method for both tasks in the MNIST dataset, resulting in improved execution accuracy for both tasks. We observe in both figures that the performance gap between the IoPm and the fixed standard Gaussian prior is larger than the gap between the IoPm and the fixed log-uniform. This reflects the larger error of the model assumption with the Gaussian distribution for the encoded feature. This demonstrates the advantage of using a learned, data-driven prior over a fixed distribution.

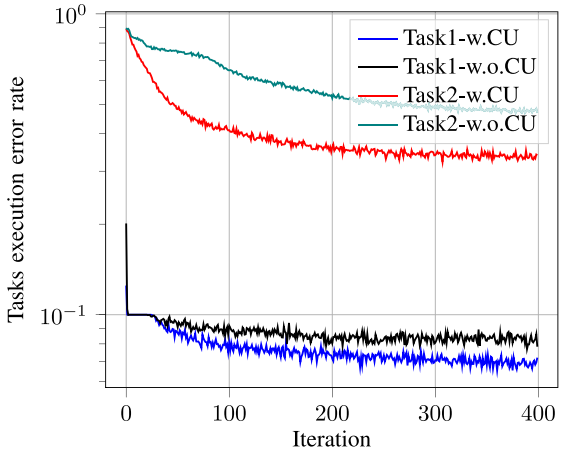
For the complex dataset, CIFAR, we observe the same performance gain for the IoPm in comparison with the standard Gaussian prior for both tasks. This is shown in Figs. 10(c) and 10(d). In addition, we note that the gap between the IoPm and the standard Gaussian grows when the task becomes more complex (as for Task2, Fig. 10(d)) and as a result its corresponding encoded feature is more complicated. Further, comparing the IoPm with the log-uniform prior, Fig. 10(d) shows that the proposed method maintains its better performance for Task2, while for the simpler one (as Task1, Fig. 10(c))the performance difference is less pronounced. This is due to the fact that the log-uniform assumption fits well in modeling the encoded feature distribution.

Overall, we observe that the proposed hybrid-learning IoPm reaches clear performance gains in comparison to the EP method. This performance gain is more considerable compared to the widely used explicit standard Gaussian prior independent of the dataset. The proposed method also mitigates the model assumption error for the latent prior in related domains, e.g., variational autoencoders, information bottleneck-based task-oriented communication, etc.

We also note that the extra computational cost introduced by the IoPm is incurred only during offline training. Therefore, the proposed method does not

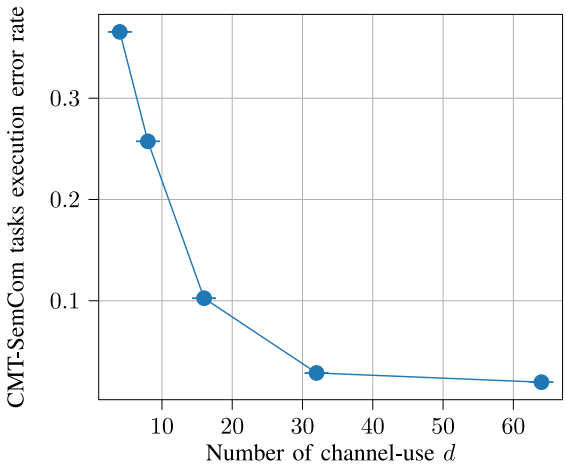


(a) Performance on MNIST dataset.

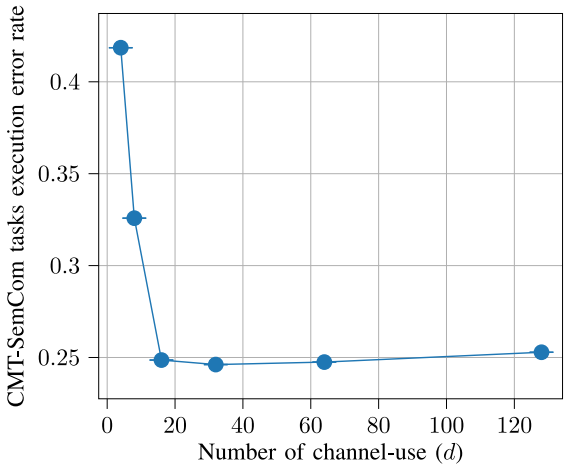


(b) Performance on CIFAR dataset.

FIGURE 8. Performance of the CMT-SemCom under the hybrid-learning IoPm.



(a) Performance on MNIST dataset.



(b) Performance on CIFAR dataset.

FIGURE 9. Impact of the rate limitation of the wireless channel on the proposed CMT-SemCom.

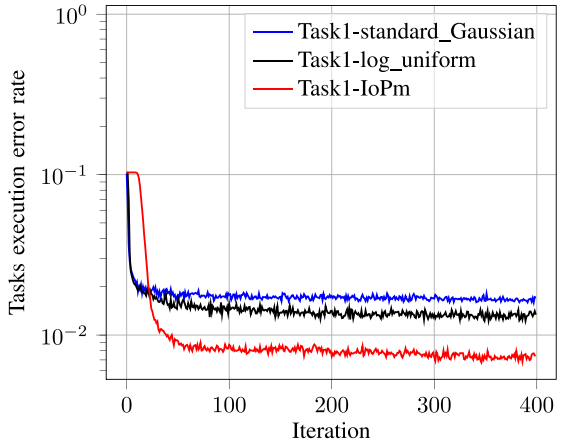
increase complexity in inference time as the DRE is used only during the training for a more accurate approximation of the KL divergence term. A detailed discussion on the comparison of computational complexity between the IoPm and the EP method is provided in Appendix E.

IV. CONCLUSION

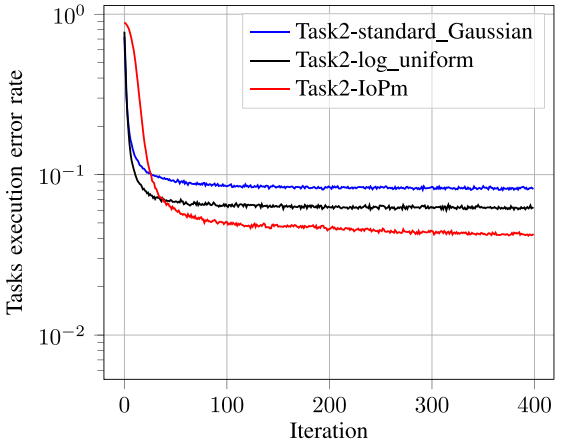
In conclusion, we advanced the CMT-SemCom framework by addressing practical constraints and extending its applicability to rate-limited wireless channels. We employed a separation-based design for the split semantic encoder to have a clear delineation of responsibilities between the CU and SUs, facilitating a more structured formulation of the communication process. Further, we tackled the regularization challenge within the joint semantic and channel coding process by employing the implicit optimal prior method (IoPm) to enhance the system's performance. We proposed a hybrid combination of DNN and kernelized-parametric ML methods to improve the approximation of the constrained problem. Through simulations on diverse datasets, we demonstrated the effectiveness of the proposed framework in achieving reliable multi-task communication under rate constraints compared to explicit prior (EP) method. Our comparisons with EP include the two widely used fixed priors in the related literature: the standard Gaussian prior and the log-uniform. These two explicit priors are widely used in the literature to model the encoded/latent feature prior. Additionally, this work brings up further research questions, such as exploring alternative parametric methods for the DRE, like the ratio matching method instead of the kernelized LR, or examining dimensionality reduction techniques to better exploit the current DRE's capabilities and improve performance in complex settings where higher dimensionality is required, i.e., more complex tasks. dynamic adaptation of semantic encoding structures to varying network requirements.

APPENDIX A
DIFFERENTIABILITY OF THE CU LOSS FUNCTION

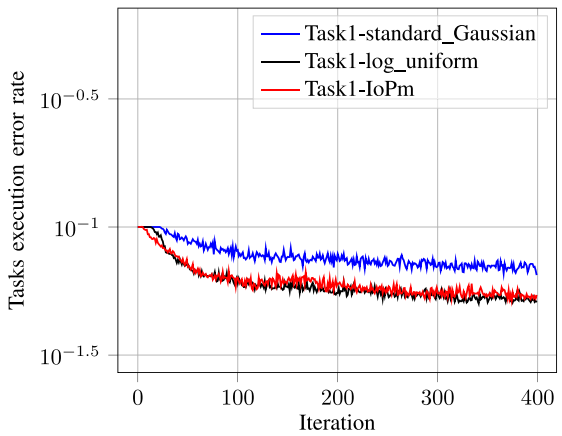
Here we first show the differentiability of (5) w.r.t $\boldsymbol{\theta}$ and then details on the reparameterization trick are provided.



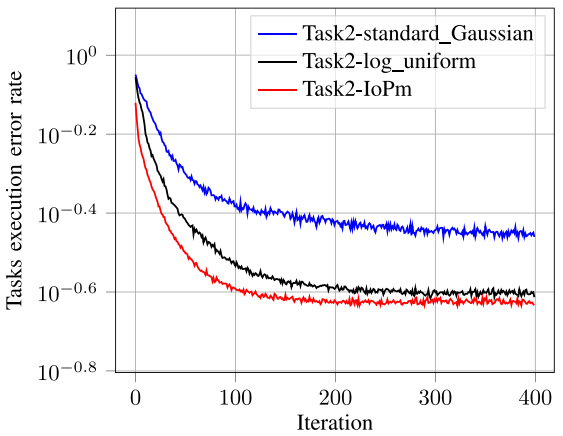
(a) Comparison for Task1 on MNIST dataset.



(b) Comparison for Task2 on MNIST dataset.



(c) Comparison for Task1 on CIFAR-10 dataset.



(d) Comparison for Task2 on CIFAR-10 dataset.

FIGURE 10. Comparison of the proposed hybrid-learning IoPm with the EP method on MNIST and CIFAR-10 datasets.

A. DERIVATIVE W.R.T THE CU ENCODER PARAMETERS

The differentiability of the CU loss function w.r.t $\boldsymbol{\Xi}$ is clear since it is explicitly stated in (5), however how $\boldsymbol{\theta}$ is updated through the backpropagation is not explicitly visible. Thus, below we show how (5) is differentiable w.r.t $\boldsymbol{\theta}$.

$$\mathcal{L}^{\mathrm{CU}}(\boldsymbol{\theta}, \boldsymbol{\Xi}) \approx \mathbb{E}_{p(\mathbf{S},\mathbf{z})}\Big[\mathbb{E}_{p_{\boldsymbol{\theta}}^{\mathrm{CU}}(\mathbf{c}|\mathbf{S})}\big[\, f(\mathbf{z})\,\big]\Big]$$

where $\mathbf{z} = g(\mathbf{c}, \boldsymbol{\Xi})$, and $\mathbf{c} = h(\mathbf{S}, \boldsymbol{\theta}, \boldsymbol{\epsilon})$. Thus, (5) can be expressed as:

$$\mathcal{L}^{\mathrm{CU}}(\boldsymbol{\theta}, \boldsymbol{\Xi}) \approx \mathbb{E}_{p(\mathbf{S},\mathbf{z})}\Big[\mathbb{E}_{p_{\boldsymbol{\theta}}^{\mathrm{CU}}(\mathbf{c}|\mathbf{S})}\big[\, f(g(h(\mathbf{S}, \boldsymbol{\theta}, \boldsymbol{\epsilon}), \boldsymbol{\Xi}))\,\big]\Big]$$

Consequently:

$$\mathcal{L}^{\mathrm{CU}}(\boldsymbol{\theta}, \boldsymbol{\Xi}) \approx f(g(h(\mathbf{S}, \boldsymbol{\theta}, \boldsymbol{\epsilon}), \boldsymbol{\Xi}))$$
$$\frac{\partial \mathcal{L}^{\mathrm{CU}}(\boldsymbol{\theta}, \boldsymbol{\Xi})}{\partial \boldsymbol{\theta}} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial h} \cdot \frac{\partial h}{\partial \boldsymbol{\theta}}$$

B. REPARAMETERIZATION TRICK

We assume that $p_{\boldsymbol{\theta}}^{\mathrm{CU}}(\mathbf{c}|\mathbf{S}) = \mathcal{N}(\mathbf{c}(\mathbf{S}; \boldsymbol{\theta}), \sigma^2\mathbf{I})$, where $\mathbf{c}(\mathbf{S}; \boldsymbol{\theta})$ states the deterministic function which maps $\mathbf{S}$ to $\mathbf{c}$ parameterized by $\boldsymbol{\theta}$. It is obvious that $\mathbf{c} \sim p_{\boldsymbol{\theta}}^{\mathrm{CU}}(\mathbf{c}|\mathbf{S})$ and then in backpropagation when the update w.r.t $\boldsymbol{\theta}$ wants to be executed there will be a problem by:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{p_{\boldsymbol{\theta}}^{\mathrm{CU}}(\mathbf{c}|\mathbf{S})}\big[\, f(g(c(\mathbf{S}; \boldsymbol{\theta}), \boldsymbol{\Xi}))\,\big]$$

Therefore, we introduce a new variable $\boldsymbol{\epsilon}$ as $\mathbf{c}_{i,l} = \mathbf{c}_i + \boldsymbol{\epsilon}_{i,l}$, where we keep $\mathbf{c}_i$ a deterministic variable and $\boldsymbol{\epsilon}_{i,l}$ a sample drawn from $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ distribution. Doing so, the expectation would be w.r.t $p(\boldsymbol{\epsilon})$ as follows, and the differentiability w.r.t $\boldsymbol{\theta}$ will be possible.

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}}\mathbb{E}_{p_{\boldsymbol{\theta}}^{\mathrm{CU}}(\mathbf{c}|\mathbf{S})}\big[\, f(g(c(\mathbf{s}; \boldsymbol{\theta}), \boldsymbol{\Xi}))\,\big] \\
&= \nabla_{\boldsymbol{\theta}}\mathbb{E}_{p(\boldsymbol{\epsilon})}\big[\, f(g(c(\boldsymbol{\epsilon}, \mathbf{S}; \boldsymbol{\theta}), \boldsymbol{\Xi}))\,\big] \\
&= \mathbb{E}_{p(\boldsymbol{\epsilon})}\big[\, \nabla_{\boldsymbol{\theta}} f(g(c(\boldsymbol{\epsilon}, \mathbf{S}; \boldsymbol{\theta}), \boldsymbol{\Xi}))\,\big] \\
&\simeq \nabla_{\boldsymbol{\theta}} f(g(c(\boldsymbol{\epsilon}, \mathbf{S}; \boldsymbol{\theta}), \boldsymbol{\Xi}))
\end{aligned}$$

APPENDIX B
THE APPROXIMATED SUs' OBJECTIVE FUNCTION

$$\begin{aligned}
\mathcal{L}^{\mathrm{SU}_n}(\boldsymbol{\phi}_n) &= I(\hat{\boldsymbol{x}}_n; z_n) - \lambda\, I(\boldsymbol{x}_n; \boldsymbol{c}) \\
&= \iint p(\hat{\boldsymbol{x}}_n, z_n)\, \log \frac{p(z_n|\hat{\boldsymbol{x}}_n)}{p(z_n)}\, dz_n\, d\hat{\boldsymbol{x}}_n
\end{aligned}$$

$$
\begin{aligned}
&- \lambda \Bigg(\iint p(\boldsymbol{x}_n, \boldsymbol{c}) \, \log \frac{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})}{p(\boldsymbol{x}_n)} \, d\boldsymbol{x}_n \, d\boldsymbol{c} \Bigg) \\
&= \Bigg[\iint p(\hat{\boldsymbol{x}}_n, z_n) \, \log p\big(z_n|\hat{\boldsymbol{x}}_n\big) \, dz_n \, d\hat{\boldsymbol{x}}_n + H(z_n) \Bigg] \\
&\quad - \lambda \Bigg(\iint p(\boldsymbol{x}_n, \boldsymbol{c}) \, \log \frac{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})}{p(\boldsymbol{x}_n)} \, d\boldsymbol{x}_n \, d\boldsymbol{c} \Bigg)
\end{aligned}
$$

Further, we omit the constant entropy term, $H(z_n)$ and exploit the underlying Markov chain structure in (1).

$$
\begin{aligned}
\mathcal{L}^{\mathrm{SU}_n}(\boldsymbol{\phi}_n) \approx & \iiiint p_{\boldsymbol{\theta}}(z_n, \boldsymbol{c}) \, p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\mathbf{c}) \\
& \qquad p\big(\hat{\boldsymbol{x}}_n|\boldsymbol{x}_n\big) \log p\big(z_n|\hat{\boldsymbol{x}}_n\big) \, dz_n \, d\hat{\boldsymbol{x}}_n \, d\boldsymbol{x}_n \, d\boldsymbol{c} \\
& - \lambda \Bigg(\iiint p_{\boldsymbol{\theta}}(z_n, \boldsymbol{c}) \\
& \qquad p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c}) \, \log \frac{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})}{p(\boldsymbol{x}_n)} \, dz_n \, d\boldsymbol{x}_n \, d\boldsymbol{c} \Bigg) \\
\approx & \, \mathbb{E}_{p_{\boldsymbol{\theta}}(z_n, \boldsymbol{c})} \Bigg[\iint p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\mathbf{c}) p\big(\hat{\boldsymbol{x}}_n|\boldsymbol{x}_n\big) \\
& \qquad \log p\big(z_n|\hat{\boldsymbol{x}}_n\big) \, d\hat{\boldsymbol{x}}_n \, d\boldsymbol{x}_n \\
& - \lambda \int p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c}) \, \log \frac{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})}{p(\boldsymbol{x}_n)} \, d\boldsymbol{x}_n \Bigg]
\end{aligned}
$$

Next, we adopt the definition of the KL [57],

$$
KL(f||g) = \int f \log \frac{f}{g},
$$

and the approximated loss function is expressed as:

$$
\begin{aligned}
&\mathcal{L}^{\mathrm{SU}_n}(\boldsymbol{\phi}_n) \\
&\approx \mathbb{E}_{p_{\boldsymbol{\theta}}(z_n, \boldsymbol{c})} \Bigg[\mathbb{E}_{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})} \Big[\mathbb{E}_{p(\hat{\boldsymbol{x}}_n|\boldsymbol{x}_n)} \big[\log p\big(z_n|\hat{\boldsymbol{x}}_n\big) \big] \Big] \\
&\qquad - \lambda \, KL\Big(p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c}) \| p(\boldsymbol{x}_n)\Big) \Bigg].
\end{aligned}
$$

It is important to note that $p_{\boldsymbol{\theta}}(z_n, \boldsymbol{c})$ is readily available at this stage, owing to the pre-trained CU. In essence, we construct our Markov chain by treating $p_{\boldsymbol{\theta}}(z_n, \boldsymbol{c})$ as a new, derived source distribution.

APPENDIX C
SU OBJECTIVE APPROXIMATION ERROR ANALYSIS

In (12), $p^{\mathrm{SU}}(\boldsymbol{x}_n|\boldsymbol{c})$, called SU encoder, is modeled in terms of a DNN parameterized by $\boldsymbol{\phi}_n$. Thus, the encoder is treated as part of model selection, and its error is about *model mismatch*. The approximation error it brings to the loss function $\mathcal{L}^{\mathrm{SU}_n}(\boldsymbol{\phi}_n, \boldsymbol{\psi}_n)$, is reflected through the KL divergence term $KL(p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c}) \| p(\boldsymbol{x}_n))$. The variational approximation error comes from the decoder, where the true posterior, $p(z_n|\hat{\boldsymbol{x}}_n)$ is approximated by $q_{\boldsymbol{\psi}_n}(z_n|\hat{\boldsymbol{x}}_n)$. The mismatch between the true and approximated posterior indicated the error in the variational approximation, and therefore, the loss will have another approximation error through the log-likelihood function (LLF).

Thus, the approximation error for the objective consists of two terms:

- LLF variational approximation error
- Model mismatch error

We begin with the LLF variational approximation error. We use a variant of Pinsker's inequality [58] which relates variational divergence to KL divergence as introduced in [59]:

$$
\sup_{A \subseteq \mathcal{Z}} |P(A) - Q(A)|^2 \leq \frac{1}{2} \, KL(P||Q),
$$

substituting the probabilities P and Q with the decoder's true and approximated ones:

$$
|p\big(z_n|\hat{\boldsymbol{x}}_n\big) - q_{\boldsymbol{\psi}_n}\big(z_n|\hat{\boldsymbol{x}}_n\big)|^2 \leq \frac{1}{2} \, KL\big(p(z_n|\hat{\boldsymbol{x}}_n)||q_{\boldsymbol{\psi}_n}(z_n|\hat{\boldsymbol{x}}_n)\big),
$$

and, similar to the definition assumed for the evidence lower bound, for instance, in variational autoencoders [59], [60], we define our approximation quality as ϵ-tight for some $\epsilon > 0$ if

$$
\mathbb{E}_{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\hat{\boldsymbol{x}}_n|\boldsymbol{c})} \big[\, KL\big(p\big(z_n|\hat{\boldsymbol{x}}_n\big)||q_{\boldsymbol{\psi}_n}\big(z_n|\hat{\boldsymbol{x}}_n\big)\big) \big] \leq \epsilon,
$$

further, we assume $p(z_n|\hat{\boldsymbol{x}}_n) \geq \alpha$ and $q_{\boldsymbol{\psi}_n}(z_n|\hat{\boldsymbol{x}}_n) \geq \alpha$.

Thus, taking the expectation from both sides of the Pinsker's inequality, we bound the variational approximation error for the LLF term as:

$$
\mathbb{E}_{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\hat{\boldsymbol{x}}_n|\boldsymbol{c})} \Big[\, |\log p\big(z_n|\hat{\boldsymbol{x}}_n\big) - \log q_{\boldsymbol{\psi}_n}\big(z_n|\hat{\boldsymbol{x}}_n\big)|^2 \Big] \leq \frac{\epsilon}{2\alpha^2} + o(\epsilon).
$$

The detailed proof of getting this bound has been provided in [59]. Next, applying the Cauchy-Schwarz inequality ($|\mathbb{E}[XY]\,|^2 \leq \mathbb{E}[X^2]\,\mathbb{E}[Y^2]$) to the error bound, we get LLF's error to our loss as follows:

$$
\begin{aligned}
&\Big| \mathbb{E}_{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\hat{\boldsymbol{x}}_n|\boldsymbol{c})} \big[\log p\big(z_n|\hat{\boldsymbol{x}}_n\big) - \log q_{\boldsymbol{\psi}_n}\big(z_n|\hat{\boldsymbol{x}}_n\big) \big] \Big| \\
&\leq \sqrt{\mathbb{E}_{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\hat{\boldsymbol{x}}_n|\boldsymbol{c})} \big[|\log p\big(z_n|\hat{\boldsymbol{x}}_n\big) - \log q_{\boldsymbol{\psi}_n}\big(z_n|\hat{\boldsymbol{x}}_n\big)|^2 \big]} \\
&\leq \sqrt{\frac{\epsilon}{2\alpha^2} + o(\epsilon)} \leq \sqrt{\frac{\epsilon}{2\alpha^2}}
\end{aligned}
$$

Next, we move to the model mismatch. We once more use the assumption of ϵ'-tightness for $\epsilon' > 0$ in the model selection. It has been shown in [61] that for the continuous case with a non-linear encoder/decoder (general neural networks), this ϵ'-tightness is satisfied. Therefore, we bound the error for our model approximation by:

$$
\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{c})} \Big[\, KL\Big(p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c}) \| p(\boldsymbol{x}_n|\boldsymbol{c})\Big) \Big] \leq \epsilon'
$$

Finally, considering $\mathcal{L}^{\mathrm{SU}_n}$ as the true objective function, the approximation error for the objective in (12) is expressed by the following upper-bound:

$$
\Big| \mathcal{L}^{\mathrm{SU}_n} - \mathcal{L}^{\mathrm{SU}_n}(\boldsymbol{\phi}_n, \boldsymbol{\psi}_n) \Big| \leq \sqrt{\frac{\epsilon}{2\alpha^2}} + \epsilon'
$$

APPENDIX D
BACKPROPAGATION OF THE SU LOSS FUNCTION

Here we show why we ignore the $\hat{r}(\mathbf{x}_n)$ in the backpropagation of the approximated objective function of the n-th SU in 20. We use the SGD over mini-batches to train our SU encoder, and the update procedure looks:

$$\boldsymbol{\phi}_n^{\tau+1} \leftarrow \boldsymbol{\phi}_n^{\tau} - \frac{\alpha}{M_\tau} \sum_{i \in M_\tau} \nabla_{\boldsymbol{\phi}_n} \mathcal{L}_i^{\mathrm{SU}_n}\left(\boldsymbol{\phi}_n^{\tau}, \boldsymbol{\psi}_n^{\tau}\right)$$

Thus, when optimized DRE is employed in the objective function, the gradient term becomes zero, and that is why we ignore the involvement of $\hat{r}(\mathbf{x}_n)$ in the optimization of the SU encoder. It is obvious that for other non-linear optimization techniques, such as Adam, the ignorance of the DRE in the updating step of the SUs is not possible.

$$\begin{aligned}\mathbb{E}_{p_{\boldsymbol{\phi}_n}(\mathbf{x}_n)}\left[\nabla_{\boldsymbol{\phi}_n} \log p_{\boldsymbol{\phi}_n}(\mathbf{x}_n)\right] &= \int p_{\boldsymbol{\phi}_n}(\mathbf{x}_n) \frac{\nabla_{\boldsymbol{\phi}_n} p_{\boldsymbol{\phi}_n}(\mathbf{x}_n)}{p_{\boldsymbol{\phi}_n}(\mathbf{x}_n)} \, d\mathbf{x}_n \\ &= \nabla_{\boldsymbol{\phi}_n} \int p_{\boldsymbol{\phi}_n}(\mathbf{x}_n) \, d\mathbf{x}_n = 0\end{aligned}$$

APPENDIX E
IoPM COMPUTATIONAL COMPLEXITY

On training computational complexity, let $\boldsymbol{m}$ be the number of the DRE training samples, and $\boldsymbol{d}$ be the dimensionality of the SU encoder output. The computational complexity of our Gaussian kernel, represented in eq. (19), will be $\mathcal{O}(m^2 d)$, which is the operations for computing the kernel Gram matrix, and therefore, $\mathcal{O}(m^2)$ for storing the matrix.

In addition, the LR classifier is trained for $\boldsymbol{T}$ iterations of gradient descent over the samples, and the total computational complexity of the training using the transformed kernel will be $\mathcal{O}(Tm^2)$. In contrast, the EP methods (fixed standard Gaussian and log-uniform priors), with which we compared our proposed IoPm, do not require the DRE and therefore have lower training complexity and training time. However, as mentioned earlier, since the DRE is present only during offline training, the training complexity is manageable due to the possibility of using powerful resources.

REFERENCES

[1] A. Halimi Razlighi, C. Bockelmann, and A. Dekorsy, "Semantic communication for cooperative multi-task processing over wireless networks," *IEEE Wireless Commun. Lett.*, vol. 13, no. 10, pp. 2867–2871, Oct. 2024.

[2] D. Gündüz et al., "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 5–41, Jan. 2023.

[3] X. Luo, H. H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 210–219, Feb. 2022.

[4] E. C. Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Comput. Netw.*, vol. 190, May 2021, Art. no. 107930. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621000773

[5] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Y. Li, "Semantic communications: Principles and challenges," 2021, *arXiv:2201.01389*.

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

[7] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.

[8] M. Sana and E. C. Strinati, "Learning semantics: An opportunity for effective 6G communications," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, 2022, pp. 631–636.

[9] D. Wheeler and B. Natarajan, "Engineering semantic communication: A survey," *IEEE Access*, vol. 11, pp. 13965–13995, 2023.

[10] W. Weaver, "Recent contributions to the mathematical theory of communication," *ETC Rev. Gen. Semantics*, vol. 10, no. 4, pp. 261–281, 1953.

[11] Y. Bar-Hillel and R. Carnap, "Semantic information," *Brit. J. Philos. Sci.*, vol. 4, no. 14, pp. 147–157, 1953. [Online]. Available: http://www.jstor.org/stable/685989

[12] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017.

[13] F. Zhou, Y. Li, X. Zhang, Q. Wu, X. Lei, and R. Q. Hu, "Cognitive semantic communication systems driven by knowledge graph," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 4860–4865.

[14] H. Xie, Z. Qin, G. Y. Li, and B. H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, 2021.

[15] H. Xie and Z. Qin, "A lite distributed semantic communication system for Internet of Things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 142–153, Jan. 2021.

[16] L. Qiao, M. B. Mashhadi, Z. Gao, C. H. Foh, P. Xiao, and M. Bennis, "Latency-aware generative semantic communications with pre-trained diffusion models," *IEEE Wireless Commun. Lett.*, vol. 13, no. 10, pp. 2652–2656, Oct. 2024.

[17] C. Xu, M. B. Mashhadi, Y. Ma, and R. Tafazolli, "Semantic-aware power allocation for generative semantic communications with foundation models," 2024, *arXiv:2407.03050*.

[18] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Y. Li, "Resource allocation for text semantic communications," *IEEE Wireless Commun. Lett.*, vol. 11, no. 7, pp. 1394–1398, Jul. 2022.

[19] Y. Wang et al., "Performance optimization for semantic communications: An attention-based reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2598–2613, Sep. 2022. [Online]. Available: https://www.ieee.org/publications/rights/index.html

[20] H. Tong, Z. Yang, S. Wang, Y. Hu, W. Saad, and C. Yin, "Federated learning based audio semantic communication over wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Madrid, Spain, 2021, pp. 1–6.

[21] Y. Wang, M. Chen, W. Saad, T. Luo, S. Cui, and H. V. Poor, "Performance optimization for semantic communications: An attention-based learning approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2021, pp. 1–6.

[22] Y. Sun et al., "Multi-functional RIS-assisted semantic anti-jamming communication and computing in integrated aerial-ground networks," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 12, pp. 3597–3617, Dec. 2024.

[23] S. Wu, H. R. Zhang, and C. Ré, "Understanding and improving information transfer in multi-task learning," 2020, *arXiv:2005.00944*.

[24] J. Shao, Y. Mao, and J. Zhang, "Learning task-oriented communication for edge inference: An information bottleneck approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 197–211, Jan. 2022.

[25] J. Shao, Y. Mao, and J. Zhang, "Task-oriented communication for multidevice cooperative edge inference," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 73–87, Jan. 2023.

[26] E. Beck, C. Bockelmann, and A. Dekorsy, "Semantic information recovery in wireless networks," *Sensors*, vol. 23, p. 6347, Jul. 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/14/6347

[27] D. Wen et al., "Task-oriented sensing, computation, and communication integration for multi-device edge ai," in *Proc. IEEE Int. Conf. Commun.*, 2023, pp. 3608–3613.

[28] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-oriented multi-user semantic communications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2584–2597, Sep. 2022.

[29] G. He, S. Cui, Y. Dai, and T. Jiang, "Learning task-oriented channel allocation for multi-agent communication," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12016–12029, Nov. 2022.

[30] Y. Sheng, F. Li, L. Liang, and S. Jin, "A multi-task semantic communication system for natural language processing," in *Proc. IEEE 96th Veh. Technol. Conf. (VTC-Fall)*, 2022, pp. 1–5.
[31] Y. E. Sagduyu, T. Erpek, A. Yener, and S. Ulukus, "Multi-receiver task-oriented communications via multi-task deep learning," in *Proc. IEEE Future Netw. World Forum (FNWF)*, 2023, pp. 1–6.
[32] M. Gong, S. Wang, and S. Bi, "A scalable multi-device semantic communication system for multi-task execution," in *Proc. IEEE Global Commun. Conf.*, 2023, pp. 2227–2232.
[33] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, pp. 41–75, Jul. 1997.
[34] A. Halimi Razlighi, M. Tillmann, E. Beck, C. Bockelmann, and A. Dekorsy, "Cooperative and collaborative multi-task semantic communication for distributed sources," 2024, *arXiv:2411.02150*.
[35] L. Yan, Z. Qin, C. Li, R. Zhang, Y. Li, and X. Tao, "QoE-based semantic-aware resource allocation for multi-task networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 11958–11971, Sep. 2024.
[36] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.
[37] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*.Cambridge, MA, USA: MIT Press, 2009.
[38] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 2. New York, NY, USA: Springer-Verlag, 2006, pp. 1122–1128.
[39] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," 2016, *arXiv:1612.00410*.
[40] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," 2000, *arXiv:physics/0004057*.
[41] S. P. Boyd and L. Vandenberghe, *Convex Optimization*.Cambridge, U.K.: Cambridge Univ. Press, 2004.
[42] J. Shao and J. Zhang, "Bottlenet++: An end-to-end approach for feature compression in device-edge co-inference systems," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2020, pp. 1–6.
[43] H. Li, J. Shao, H. He, S. Song, J. Zhang, and K. B. Letaief, "Tackling distribution shifts in task-oriented communication with information bottleneck," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 7, pp. 2667–2683, Jul. 2025.
[44] F. Binucci, P. Banelli, P. Di Lorenzo, and S. Barbarossa, "Opportunistic information-bottleneck for goal-oriented feature extraction and communication," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 2418–2432, 2024.
[45] S. Xie, S. Ma, M. Ding, Y. Shi, M. Tang, and Y. Wu, "Robust information bottleneck for task-oriented communication with digital modulation," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 8, pp. 2577–2591, Aug. 2023.
[46] H. Takahashi, T. Iwata, Y. Yamanaka, M. Yamada, and S. Yagi, "Variational autoencoder with implicit optimal priors," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5066–5073.
[47] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density Ratio Estimation in Machine Learning*.Cambridge, U.K.: Cambridge Univ. Press, 2012.
[48] D. P. Bertsekas, "Nonlinear programming," *J. Oper. Res. Soc.*, vol. 48, no. 3, pp. 334–334, 1997.
[49] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.
[50] D. Molchanov, A. Ashukha, and D. Vetrov, "Variational dropout sparsifies deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2498–2507.
[51] Z. Lyu, G. Zhu, J. Xu, B. Ai, and S. Cui, "Semantic communications for image recovery and classification via deep joint source and channel coding," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 8388–8404, Aug. 2024.
[52] J. Pei, C. Feng, P. Wang, H. Tabassum, and D. Shi, "Latent diffusion model-enabled low-latency semantic communication in the presence of semantic ambiguities and wireless channel noises," *IEEE Trans. Wireless Commun.*, vol. 24, no. 5, pp. 4055–4072, May 2025.
[53] S. Ma et al., "Task-oriented explainable semantic communications," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9248–9262, Dec. 2023.
[54] C. Cai, X. Yuan, and Y.-J. Angela Zhang, "Multi-device task-oriented communication via maximal coding rate reduction," *IEEE Trans. Wireless Commun.*, vol. 23, no. 12, pp. 18096–18110, Dec. 2024.
[55] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[56] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.
[57] T. M. Cover, *Elements of Information Theory*.Hoboken, NJ, USA: Wiley, 1999.
[58] M. S. Pinsker, *Information and Information Stability of Random Variables and Processes*. San Francisco, CA, USA: Holden-Day, 1964.
[59] A. Shekhovtsov, D. Schlesinger, and B. Flach, "VAE approximation error: ELBO and exponential families," 2021, *arXiv:2102.09310*.
[60] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends® Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.
[61] B. Dai and D. Wipf, "Diagnosing and enhancing VAE models," 2019, *arXiv:1903.05789*.

A. HALIMI RAZLIGHI (Graduate Student Member, IEEE) received the B.Sc. degree (top graduate student) in electrical engineering-telecommunication from Iran Broadcasting University (IRIBU), Tehran, Iran, in 2018, and the M.Sc. degree in electrical engineering-communication-systems Iran the University of Science and Technology, Tehran, in 2022. He is currently pursuing the Ph.D. degree in communication engineering with the Department of Communications Engineering, University of Bremen, Germany. His research interests include semantic communication, task-oriented communication, and machine learning for wireless communication.

CARSTEN BOCKELMANN (Member, IEEE) received the Dipl.-Ing. and Ph.D. degrees in electrical engineering from the University of Bremen, Germany in 2006 and 2012, respectively. Since 2012, he has been with the Department of Communications Engineering (ANT), University of Bremen as a Senior Research Group Leader. His main research interests include but are not limited to applications of compressive sensing and machine learning in communication, massive machine-type communication, ultra-reliable low latency communication, compressive sampling, and channel coding.

ARMIN DEKORSY (Senior Member, IEEE) is a Professor with the University of Bremen, where he is the Director of the Gauss-Olbers Space Technology Transfer Center and Heads the Department of Communications Engineering. With over 11 years of industry experience, including distinguished research positions, such as a DMTS with Bell Labs and a Research Coordinator Europe with Qualcomm, he has actively participated in more than 65 international research projects, with leadership roles in 17 of them. He co-authored the textbook *Nachrichtenübertragung* (Release 6, Springer Vieweg), which is a bestseller in the field of communication technologies in Germanspeaking countries. His research focuses on signal processing and wireless communications for 5G/6G, industrial radio, and 3-D networks. He is a Senior Member of the IEEE Communications and Signal Processing Society and a member of the VDE/ITG Expert Committee on Information and System Theory.