# Semantic Communication for Cooperative Multi-Tasking over Rate-Limited Wireless Channels with Implicit Optimal Prior

Ahmad Halimi Razlighi, Carsten Bockelmann, and Armin Dekorsy
Department of Communications Engineering, University of Bremen, Germany
E-mails:{halimi, bockelmann, dekorsy}@ant.uni-bremen.de

***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*—Semantic communication, cooperative multi-tasking, information theory, implicit optimal prior, deep learning, parametric methods, hybrid learning.**

## 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]–[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 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 communica-

This work was supported by the German Ministry of Education and Research (BMBF) under Grant 16KISK016 (Open6GGub).

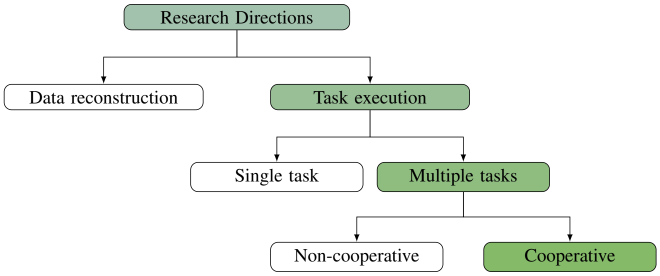


Fig. 1: Categorization of research works in semantic communications.

tion domain, relying on defined loss functions and the black-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]–[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].

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 the I-A.

### *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, [22] developed a communication scheme using the information bottleneck (IB) framework, which encodes information while adapting to dynamic channel conditions. Moreover, the same authors in [23] studied distributed relevant information encoding for collaborative feature extraction to fulfill a single task. [24] also offered a framework for collaborative retrieval of the message using multiple received semantic information.

To address practical communication scenarios, SemCom systems must be capable of handling multiple tasks simultaneously. Early efforts, such as [25], [26], explored non-cooperative methods where each task operates on its respective dataset independently. Conversely, recent works like [27]–[29] studied joint multi-tasking using established ML approaches and architecture [30] 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 black-box use of DNN. [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, [31] 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 [31], 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, [32] has studied resource allocation for multi-task SemCom networks.

### *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], [31], 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., [22], [33], 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.

TABLE I: 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 |
| $\epsilon$ | 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 |
| $\sigma_k$ | kernel bandwidth |

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 II-B and II-C, respectively. Next, 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 I.

## 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 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 II-B, while employing the information bottleneck (IB) approach in II-C to formulate the objective function for our constrained optimization problem. Next, II-D presents the IoPm and our hybrid learning approach.

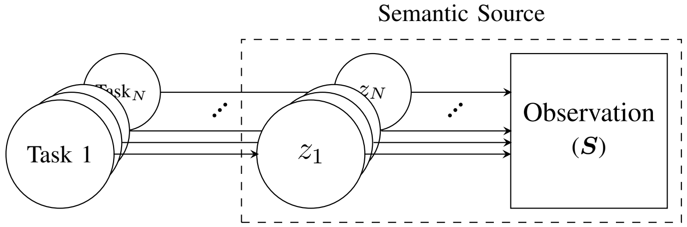


Fig. 2: Probabilistic graphical modeling of the semantic source.

### 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 $\boldsymbol{z} = [z_1\, z_2\, \dots\, z_N]$. 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 [34] and a stack view for a better illustration. Such a definition enables the simultaneous extraction of multiple semantic variables based on a single observation.

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 $p^{\text{CU}}(\boldsymbol{c}|\boldsymbol{S})$. Next, each SU$_n$ encodes $\boldsymbol{c}$ into a task-specific information $\boldsymbol{x}_n$ using $p^{\text{SU}_n}(\boldsymbol{x}_n|\boldsymbol{c})$. 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 $p^{\text{Dec}_n}(\hat{z}_n|\hat{\boldsymbol{x}}_n)$ 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:

$$\begin{aligned} & p(\hat{z}_n, \hat{\boldsymbol{x}}_n, \boldsymbol{x}_n, \boldsymbol{c}|\boldsymbol{S}) = \\ & p^{\text{Dec}_n}(\hat{z}_n|\hat{\boldsymbol{x}}_n)\, p^{\text{Channel}}(\hat{\boldsymbol{x}}_n|\boldsymbol{x}_n)\, p^{\text{SU}_n}(\boldsymbol{x}_n|\boldsymbol{c}) p^{\text{CU}}(\boldsymbol{c}|\boldsymbol{S}). \end{aligned} \tag{1}$$

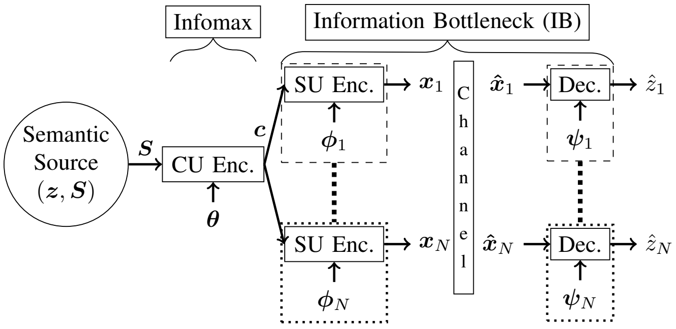


Fig. 3: Illustration of the proposed separation-based design for the CMT-SemCom framework under rate-limit wireless channels.

### 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.

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

Hence, our objective is to maximize the mutual information between the CU output,$\boldsymbol{c}$, and the underlying semantic variables, $\boldsymbol{z} = [\, z_1 \, z_2 \, \dots \, z_N]^T$, 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 [35]. 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) [33]. The technique is widely used in machine learning, e.g., [36], and also in task-oriented communications, e.g., [22] and [23]. Thus, we approximate the posterior distribution $p^{\text{CU}}(\boldsymbol{c}|\boldsymbol{S})$ by variational approximation using NN parameterized by $\boldsymbol{\theta}$. This approximation yields $p^{\text{CU}}_{\boldsymbol{\theta}}(\boldsymbol{c}|\boldsymbol{S})$ and we present the CU objective function as follows.

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

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 $p(\boldsymbol{S}, \boldsymbol{z})$ and the posterior distribution $p^{\text{CU}}_{\boldsymbol{\theta}}(\boldsymbol{c}|\boldsymbol{S})$, the semantic space posterior $p(\boldsymbol{z}|\boldsymbol{c})$ could be fully determined:

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

Considering that $p_{\boldsymbol{\theta}}(\boldsymbol{c}) = \int p^{\text{CU}}_{\boldsymbol{\theta}}(\boldsymbol{c}|\boldsymbol{S})\, p(\boldsymbol{S}) \, d\boldsymbol{S}$ 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 $\boldsymbol{\Xi} = [\, \boldsymbol{\xi}_1 \, \boldsymbol{\xi}_2 \, \dots \, \boldsymbol{\xi}_N]^T$ 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.

**Algorithm 1** Training the CU Encoder.

```
Input: Preprocessed training dataset: $\{\boldsymbol{S}^{(m)}, z_1^{(m)}, \dots, z_N^{(m)}\}_{m \in M}$
    number of iterations $T$, batch sizes $M_n$
Output: The trained parameters $\boldsymbol{\theta}$ and $\Xi$
for epoch $t = 1$ to $T$
    for $n = 1$ to $N$
        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 $M_n$
            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 $\Xi$ through backpropagation.
end for
```

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

Further, we approximate the expectations with Monte Carlo sampling following data-driven approach, given that there exists a dataset $\{\mathbf{S}^{(m)}, z_1^{(m)}, \dots, z_N^{(m)}\}_{m=1}^{M}$ 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:

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

Additionally, we have applied the reparameterization trick [37], to overcome the differentiability issues in the backpropagation of the objective function by introducing $\boldsymbol{c}^{(m,l)} = \boldsymbol{\mu}_{\mathbf{c}|\boldsymbol{S}^m} + \boldsymbol{\sigma}_{\mathbf{c}|\boldsymbol{S}^m} \odot \boldsymbol{\epsilon}^l$ 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.

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

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) [38], 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 [39] to optimization problem (7), we reformulate the objective function as:

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

We drop the constant term, $\lambda R_n$, 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, $p^{\mathrm{SU}_n}(\boldsymbol{x}_n|\boldsymbol{c})$, with NN parameterized by $\boldsymbol{\phi}_n$, yielding $p^{\mathrm{SU}_n}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})$, we end up with the following objective function:

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

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:

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

As we have a DNN-based implementation followed by an E2E learning fashion, improved to be effective for task-oriented communication [40], 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.

For the regularization term in (12), where the KL divergence appears, adopting a variational marginal posterior distribution for $p_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n)$, 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 a standard Gaussian distribution $\mathcal{N}(\mathbf{0},\mathbf{I})$ has been the most common approach taken in the literature so that the KL term can be calculated in closed form. This approach was introduced and mostly used in training the variational autoencoder structures [37]. However, fixing the prior distribution either to a standard Gaussian or any other distribution is a sub-optimal measure. Hence, we slightly change the loss function to leverage density ratio estimation in order to optimally estimate the KL divergence and consequently, to better estimate the objective function.

### D. Implicit Optimal Prior Method

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

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

where $q(\boldsymbol{x}_n)$ is a standard Gaussian distribution that causes the KL divergence to be calculated in a closed form. This trick, introduced in [41] for variational auto-encoders, enables us to implicitly manage the prior distribution by estimating the density ratio $p_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n)/q(\boldsymbol{x}_n)$ and prevent fixing a distribution. [41] 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 [41] to the 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 implicit optimal prior method (IoPm) in CMT-SemCom, we follow the *probabilistic classification* approach amongst other approaches of density ratio estimation [42]. 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 Gaussian prior distribution, $q(\boldsymbol{x}_n)$, and samples drawn from the distribution produced by the semantic encoder, $p_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n)$. The key insight is

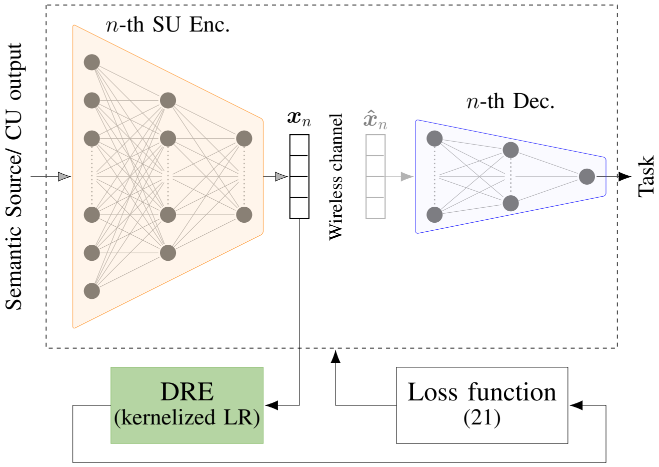


Fig. 4: Illustration of the proposed hybrid learning approach for the $n$-th SU.

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 $p_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n)$ which are available using ancestral sampling [35] from the output of our encoder, $p^{\text{SU}}_{\boldsymbol{\phi}_n}(\mathbf{x}_n|\mathbf{c})$. Then, $p(\boldsymbol{x}_n|y)$ is defined as:

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

Thus, our density ratio can be expressed as:

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

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:

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

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:

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

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 [43]. Finally, using (15) and (16), our density ratio estimator (DRE) is expressed as:

**Algorithm 2** Density Ratio Estimation using Kernelized LR

```
Input: Samples $\boldsymbol{x}_p \sim p_{\boldsymbol{\phi}_n}(\mathbf{x}_n)$, samples $\boldsymbol{x}_q \sim q(\mathbf{x}_n)$, kernel bandwidth $\sigma_k$
Output: 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 $K_{\text{new}}$ between saved $\boldsymbol{X}'$ from training and the new data $\boldsymbol{X}$
$\hat{r}(\boldsymbol{x}_{\boldsymbol{n}}) = \exp(K_{\text{new}} \cdot \hat{\boldsymbol{\omega}})$
Step 6: Return estimated density ratio $\hat{r}(\boldsymbol{x})$
```

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

For the DRE in (18), we use the *Gaussian kernel* for the basis function with kernel bandwidth, $\sigma_k$ 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 [35] 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.

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:

$$\mathcal{L}^{\text{SU}_n}(\boldsymbol{\phi}_n,\boldsymbol{\psi}_n) \approx$$

$$\mathbb{E}_{p_{\boldsymbol{\theta}}(z_n,\boldsymbol{c})}\Big[\ \mathbb{E}_{p^{\text{SU}}_{\boldsymbol{\phi}_n}(\hat{\boldsymbol{x}}_n|\boldsymbol{c})}\left[\log q_{\boldsymbol{\psi}_n}(z_n|\hat{\boldsymbol{x}}_n)\right]$$

$$-\ \lambda\ \Big\{KL(p^{\text{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})\parallel q(\boldsymbol{x}_n)) - \mathbb{E}_{p^{\text{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})}\left[\ \log\hat{r}(\boldsymbol{x}_n)\right]\ \Big\}\Big]\ . \tag{20}$$

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:

$$\mathcal{L}^{\text{SU}_n}(\boldsymbol{\phi}_n,\boldsymbol{\psi}_n) \approx \frac{1}{M_n}\sum_{m=1}^{M_n}\left\{\frac{1}{L}\sum_{l=1}^{L}\left[\log q_{\boldsymbol{\psi}_n}(z_n|\hat{\boldsymbol{x}}_n^{(m,l)})\right]\right.$$

$$\left. -\ \lambda\left[KL(p^{\text{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n^{(m)}|\boldsymbol{c})\parallel q(\boldsymbol{x}_n)) - \frac{1}{L}\sum_{l=1}^{L}\hat{r}(\boldsymbol{x}_n^{(m,l)})\right]\right\}. \tag{21}$$

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, $\mathbb{E}_{p^{\text{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})}[\ \log\hat{r}(\boldsymbol{x}_n)]$, 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 C.

*1) KL Divergence closed-Form Expression:* Since we assumed the Gaussian distribution for our SU encoder such that $p^{\text{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}_n|\boldsymbol{c}^m},\boldsymbol{\sigma}_{\boldsymbol{x}_n|\boldsymbol{c}^m}\mathbf{I})$, the KL term in (21) can be decomposed into a summation, leveraging $p^{\text{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c}) = \prod_i^{N_x} p^{\text{SU}}_{\boldsymbol{\phi}_n}(x_n^{(i)}|\boldsymbol{c})$ as $\boldsymbol{x}_n \in \mathbb{R}^{N_x\times 1}$.

$$\begin{aligned} &KL(p^{\text{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})\ \|q(\boldsymbol{x}_n)) \\ &\qquad = \sum_{i=1}^{N_x} KL(p^{\text{SU}}_{\boldsymbol{\phi}_n}(x_n^{(i)}|\boldsymbol{c})\parallel q(x_n^{(i)})). \end{aligned} \tag{22}$$

Then, we utilize the closed-form expression for the KL between two Gaussian distributions:

$$\begin{aligned} &KL(p^{\text{SU}}_{\boldsymbol{\phi}_n}(x_n^{(i)}|\boldsymbol{c})\parallel q(x_n^{(i)})) \\ &\qquad = \frac{1}{2}\left(\log\frac{1}{\sigma^2_{x_n|\boldsymbol{c}}} + \sigma^2_{x_n|\boldsymbol{c}} + \mu^2_{x_n|\boldsymbol{c}} - 1\right)\ . \end{aligned} \tag{23}$$

**Algorithm 3** Training the $n$-th SU Encoder and Decoder.

```
Input: 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 $M_n$
Output: The trained parameters $\boldsymbol{\phi}_n, \boldsymbol{\psi}_n$, and $\boldsymbol{\omega}$
for epoch $t = 1$ to $T$
    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 $p^{\text{CU}}_{\boldsymbol{\theta}}(\boldsymbol{c}|\boldsymbol{S})$
    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 $M_n$
        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
```

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

## III. Simulation Results

To evaluate the effectiveness of our proposed separation-based CMT-SemCom design over rate-limited wireless channels using the hybrid learning framework, we consider two representative tasks: 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. Additionally, we examine system behavior under different channel constraint levels. Finally, we compare the IoPm with a baseline approach that uses an explicit fixed prior (EP) [1].

### A. Simulation Setup

We examine the proposed framework across two famous datasets. The MNIST dataset of handwritten digits [44], contains 60,000 images for the training set and 10,000 samples for the test set. Moreover, the CIFAR-10 dataset [45] 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 $\{\boldsymbol{S}^{(m)}, z_1^{(m)}, ..., z_T^{(m)}\}_{m=1}^{M}$. The implemented DNN structure and the specification of the parameterized DRE are listed in Table II. The specifications are found heuristically such that the performance is maximized.

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

TABLE II: Encoder-decoder NN architecture for the proposed CMT-SemCom along with the specification of the parametric-based kernelized DRE.

(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 ($\mathrm{T}_1$) | size: 1, activation: Sigmoid |
| | Dense ($\mathrm{T}_2$) | size: 10, activation: Softmax |
| **SU** | Dense | size: 64, activation: ReLU |
| | Dense | size: 64, activation: ReLU |
| | Dense ($\boldsymbol{\mu_{x_n}}$) | size: 32, activation: Tanh |
| | Dense ($\boldsymbol{\sigma_{x_n}}$) | size: 32, activation: Sigmoid |
| **Dec** | Dense | size: 32, activation: ReLU |
| | Dense ($\mathrm{T}_1$) | size: 1, activation: Sigmoid |
| | Dense ($\mathrm{T}_2$) | 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 ($\mathrm{T}_1$) | size: 1, activation: Sigmoid |
| | Dense ($\mathrm{T}_2$) | size: 10, activation: Softmax |
| **SU** | Dense | size: 256, activation: ReLU |
| | Dense | size: 256, activation: ReLU |
| | Dense ($\boldsymbol{\mu_{x_n}}$) | size: 128, activation: Tanh |
| | Dense ($\boldsymbol{\sigma_{x_n}}$) | size: 128, activation: Sigmoid |
| **Dec** | Dense | size: 128, activation: ReLU |
| | Dense ($\mathrm{T}_1$) | size: 1, activation: Sigmoid |
| | Dense ($\mathrm{T}_2$) | size: 10, activation: Softmax |

(c) Specification of the 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 |

### 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., [41], 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.

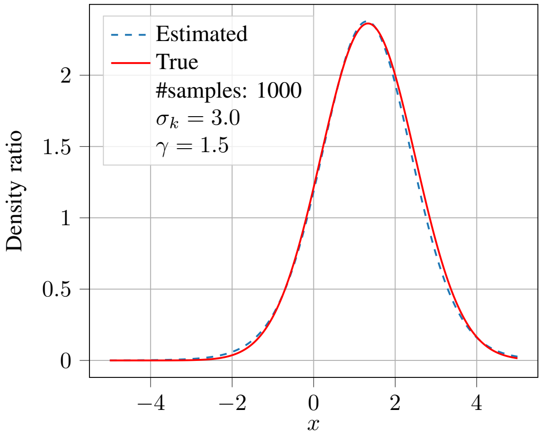
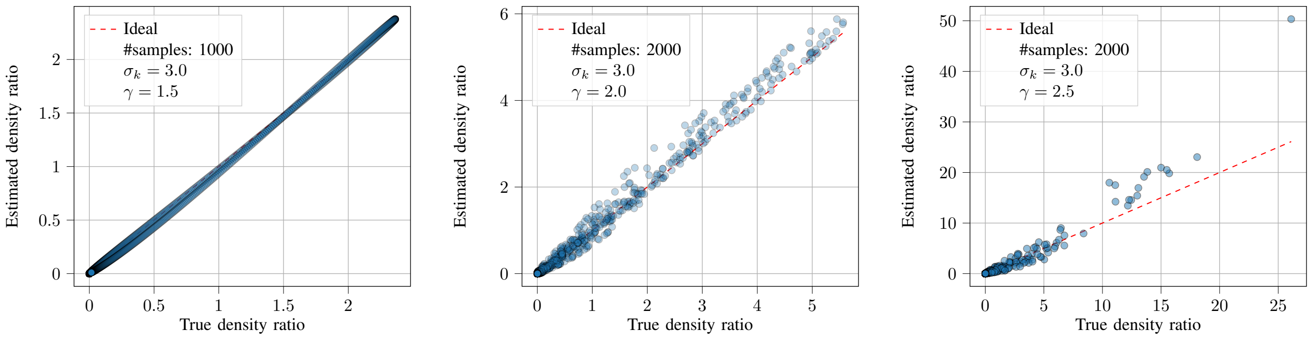


Fig. 5: Performance of the proposed DRE for the scalar data.

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 cases with increasing dimensions. Fig. 6 presents scatter plots of the estimated density ratios in 1D, 2D, and 4D. The evaluations in Fig. 6, 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. 7, 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



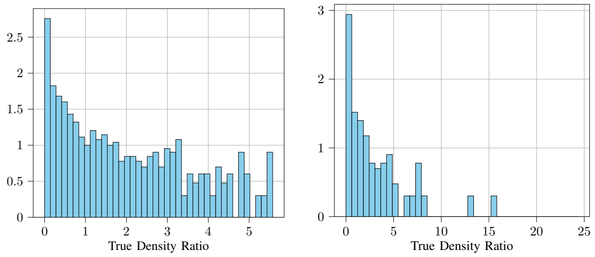
(a) Performance in 1D

(b) Performance in 2D

(c) Performance in 4D

Fig. 6: Performance of the proposed DRE for different data dimensions.

estimator to distinguish between regions of high and low density, especially under limited sampling.



(a) 2D

(b) 4D

Fig. 7: The histogram of the true density ratios for different data dimensions.

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 II 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): Each SU execute its task independently, using the semantic source directly as input without any cooperation.

Fig. 8 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.

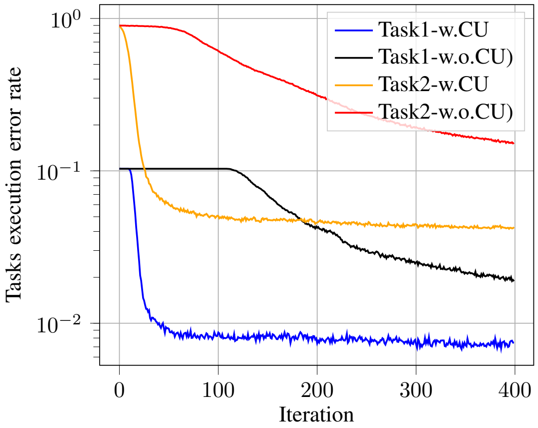


Fig. 8: Performance of the CMT-SemCom under the hybrid learning approach and the IoPm for the MNIST dataset.

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.

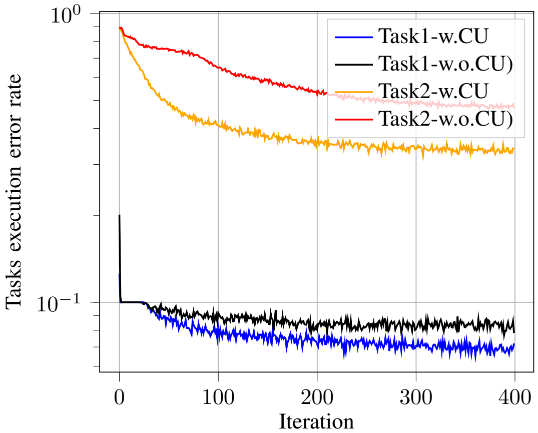


Fig. 9: Performance of the CMT-SemCom under the hybrid learning approach and the IoPm for the CIFAR dataset.

A similar behavior is observed for the CIFAR dataset, as

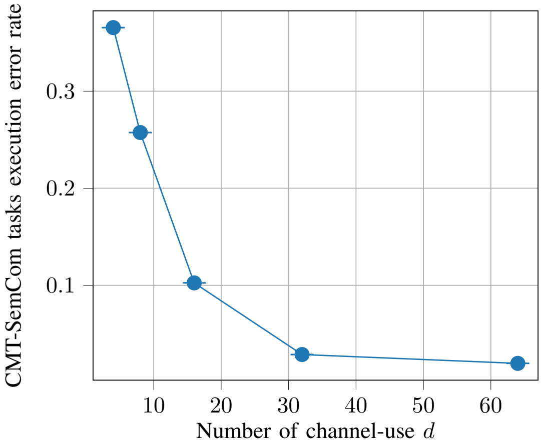


Fig. 10: Impact of the rate limitation of the wireless channel on CMT-SemCom performance for the MNIST dataset.

shown in Fig. 9. 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 available channel uses ($d$). The average task execution error rate for both tasks serves as the evaluation metric to capture how constrained bandwidth affects system accuracy.

Fig. 10 and Fig. 11 illustrate this effect for the MNIST and CIFAR datasets, respectively.

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.

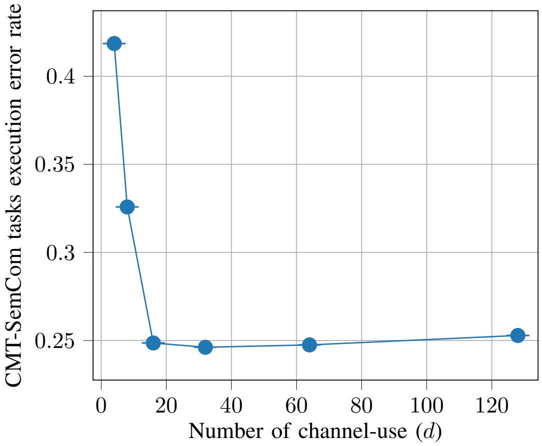


Fig. 11: Impact of the rate limitation of the wireless channel on CMT-SemCom performance for the CIFAR dataset.

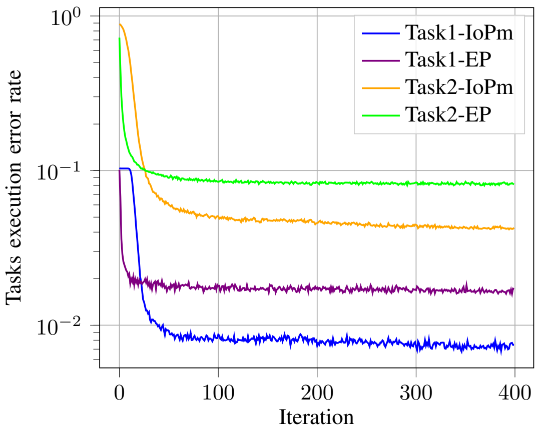


Fig. 12: Comparing the proposed hybrid-based IoPm with the EP for the MNIST dataset.

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-Based IoPm Vs. EP

Finally, we compare our hybrid-based IoPm approach with the commonly used EP method, which employs a fixed standard Gaussian distribution for the prior. As shown in Fig. 12, our hybrid-based IoPm consistently outperforms the EP method in the MNIST dataset. resulting in improved execution accuracy for both tasks. This demonstrates the advantage of using a learned, data-driven prior over a fixed distribution.

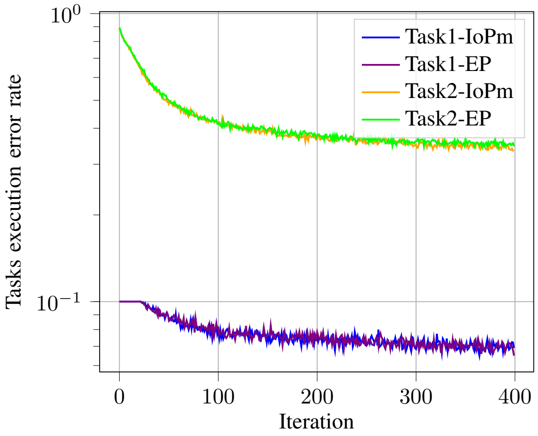


Fig. 13: Comparing the proposed hybrid-based IoPm with the EP for the CIFAR dataset.

For the complex dataset, CIFAR, as shown in Fig. 13, the performance difference between the proposed IoPm and the EP method is less pronounced. In fact, for both tasks, the IoPm does not significantly outperform EP. While IoPm maintains competitive performance in the simpler binary classification task (Task1), the results are largely comparable to EP in the more complex task (Task2). This suggests that while the hybrid-based IoPm remains competitive, its advantage over the EP method diminishes as dataset complexity increases.

Overall, we observe that the proposed hybrid-based IoPm reaches clear performance gains in less complex datasets, while offering comparable results in more complex ones. These findings motivate further research into enhancing the DRE unit. Future investigations could explore alternative parametric methods for the DRE, such as 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.

## 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 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, particularly for less complex datasets. Additionally, this work brings up further research questions, such as exploring other methods for better performance in dealing with complex datasets and 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. Derivative w.r.t the CU Encoder Parameters

The differentiability of the CU loss function w.r.t $\Xi$ 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 $\boldsymbol{\theta}$.

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

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

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

Consequently:

$$\mathcal{L}^{\text{CU}}(\boldsymbol{\theta}, \Xi) \approx f(g(h(\mathbf{S}, \boldsymbol{\theta}, \boldsymbol{\epsilon}), \Xi))$$

$$\frac{\partial \mathcal{L}^{\text{CU}}(\boldsymbol{\theta}, \Xi)}{\partial \boldsymbol{\theta}} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial h} \cdot \frac{\partial h}{\partial \boldsymbol{\theta}}$$

### B. Reparameterization Trick

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

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

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.

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

## Appendix B
## The Approximated SUs' Objective Function

$$\begin{aligned}
\mathcal{L}^{\text{SU}_n}(\boldsymbol{\phi}_n) &= I(\hat{\boldsymbol{x}}_n; z_n) - \lambda\, I(\boldsymbol{x}_n; \boldsymbol{c}) \\
&= \iint p(\hat{\boldsymbol{x}}_n, z_n)\, \log \frac{p(z_n|\hat{\boldsymbol{x}}_n)}{p(z_n)}\, dz_n\, d\hat{\boldsymbol{x}}_n \\
&\quad - \lambda\Big(\iint p(\boldsymbol{x}_n, \boldsymbol{c})\, \log \frac{p_{\boldsymbol{\phi}_n}^{\text{SU}}(\boldsymbol{x}_n|\boldsymbol{c})}{p(\boldsymbol{x}_n)}\, d\boldsymbol{x}_n\, d\boldsymbol{c}\Big) \\
&= \Big[\iint p(\hat{\boldsymbol{x}}_n, z_n)\, \log p(z_n|\hat{\boldsymbol{x}}_n)\, dz_n\, d\hat{\boldsymbol{x}}_n + H(z_n)\Big] \\
&\quad - \lambda\Big(\iint p(\boldsymbol{x}_n, \boldsymbol{c})\, \log \frac{p_{\boldsymbol{\phi}_n}^{\text{SU}}(\boldsymbol{x}_n|\boldsymbol{c})}{p(\boldsymbol{x}_n)}\, d\boldsymbol{x}_n\, d\boldsymbol{c}\Big)
\end{aligned}$$

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

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

Next, we adopt the definition of the KL [46],

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

in addition to adopting the notation $p_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n)$ for the marginal output distribution of the $n$-th SU to get the approximated loss function below:

$$\mathcal{L}^{\mathrm{SU}_n}(\boldsymbol{\phi}_n) \approx \mathbb{E}_{p_{\boldsymbol{\theta}}(z_n,\boldsymbol{c})}\Big[\, \mathbb{E}_{p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c})}\big[\mathbb{E}_{p(\hat{\boldsymbol{x}}_n|\boldsymbol{x}_n)}[\, \log p(z_n|\hat{\boldsymbol{x}}_n)\,]\,\big]$$

$$- \lambda\, KL(p^{\mathrm{SU}}_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n|\boldsymbol{c}) \parallel p_{\boldsymbol{\phi}_n}(\boldsymbol{x}_n))\Big] \, .$$

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
## Backpropagation of the SU Loss Function

Here we show why we ignore the $\hat{r}(\mathbf{x}_n)$ in the backpropagation of our 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:

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

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.

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

$$= \nabla_{\boldsymbol{\phi}_n} \int p_{\boldsymbol{\phi}_n}(\mathbf{x}_n)\, d\mathbf{x}_n = 0$$

## References

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

[2] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, “Beyond transmitting bits: Context, semantics, and task-oriented communications,” *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 5–41, 2023.

[3] X. Luo, H. H. Chen, and Q. Guo, “Semantic communications: Overview, open issues, and future research directions,” *IEEE Wireless Communications*, vol. 29, pp. 210–219, 2 2022.

[4] E. Calvanese Strinati and S. Barbarossa, “6G networks: Beyond shannon towards semantic and goal-oriented communications,” *Computer Networks*, vol. 190, p. 107930, 2021. [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,” *arXiv preprint arXiv:2201.01389*, 2021.

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

[7] C. E. Shannon, “A mathematical theory of communication,” 1948.

[8] M. Sana and E. C. Strinati, “Learning semantics: An opportunity for effective 6G communications.” Institute of Electrical and Electronics Engineers Inc., 2022, pp. 631–636.

[9] D. Wheeler and B. Natarajan, “Engineering semantic communication: A survey,” *IEEE Access*, vol. 11, pp. 13 965–13 995, 2023.

[10] W. Weaver, “Recent contributions to the mathematical theory of communication,” *ETC: a review of general semantics*, pp. 261–281, 1953.

[11] Y. Bar-Hillel and R. Carnap, “Semantic information,” *The British Journal for the Philosophy of Science*, 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 Transactions on Knowledge and Data Engineering*, vol. 29, pp. 2724–2743, 12 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 *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 4860–4865.

[14] H. Xie, Z. Qin, G. Y. Li, and B. H. Juang, “Deep learning enabled semantic communication systems,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.

[15] H. Xie and Z. Qin, “A lite distributed semantic communication system for internet of things,” *IEEE Journal on Selected Areas in Communications*, vol. 39, pp. 142–153, 1 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 Communications Letters*, vol. 13, no. 10, pp. 2652–2656, 2024.

[17] C. Xu, M. B. Mashhadi, Y. Ma, and R. Tafazolli, “Semantic-aware power allocation for generative semantic communications with foundation models,” 2024. [Online]. Available: https://arxiv.org/abs/2407.03050

[18] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Y. Li, “Resource allocation for text semantic communications,” *IEEE Wireless Communications Letters*, vol. 11, pp. 1394–1398, 7 2022.

[19] Y. Wang, S. Member, M. Chen, T. Luo, S. Member, W. Saad, D. Niyato, H. V. Poor, L. Fellow, S. Cui, and P. Cheng, “Performance optimization for semantic communications: An attention-based reinforcement learning approach and also with the,” *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 40, 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.” Institute of Electrical and Electronics Engineers Inc., 2021.

[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 *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[22] 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, 2022.

[23] J. Shao, Y. Mao, and J. Zhang, “Task-oriented communication for multidevice cooperative edge inference,” *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 73–87, 2023.

[24] E. Beck, C. Bockelmann, and A. Dekorsy, “Semantic information recovery in wireless networks,” *Sensors*, vol. 23, p. 6347, 7 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/14/6347

[25] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, “Task-oriented multiuser semantic communications,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2584–2597, 2022.

[26] G. He, S. Cui, Y. Dai, and T. Jiang, “Learning task-oriented channel allocation for multi-agent communication,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 12 016–12 029, 2022.

[27] Y. Sheng, F. Li, L. Liang, and S. Jin, “A multi-task semantic communication system for natural language processing,” in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, 2022, pp. 1–5.

[28] Y. E. Sagduyu, T. Erpek, A. Yener, and S. Ulukus, “Multi - receiver task-oriented communications via multi - task deep learning,” in *2023 IEEE Future Networks World Forum (FNWF)*, 2023, pp. 1–6.

[29] M. Gong, S. Wang, and S. Bi, “A scalable multi-device semantic communication system for multi-task execution,” in *GLOBECOM 2023-2023 IEEE Global Communications Conference*. IEEE, 2023, pp. 2227–2232.

[30] R. Caruana, “Multitask learning,” *Machine learning*, vol. 28, pp. 41–75, 1997.

[31] A. Halimi Razlighi, M. Tillmann, E. Beck, C. Bockelmann, and A. Dekorsy, “Cooperative and collaborative multi-task semantic communication for distributed sources,” 2024. [Online]. Available: https://arxiv.org/abs/2411.02150

[32] L. Yan, Z. Qin, C. Li, R. Zhang, Y. Li, and X. Tao, “Qoe-based semantic-aware resource allocation for multi-task networks,” *IEEE Transactions on Wireless Communications*, vol. 23, no. 9, pp. 11 958–11 971, 2024.

[33] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[34] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[35] C. M. Bishop, "Pattern recognition and machine learning," *Springer google schola*, vol. 2, pp. 1122–1128, 2006.

[36] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[37] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.

[38] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," 2000.

[39] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[40] J. Shao and J. Zhang, "Bottlenet++: An end-to-end approach for feature compression in device-edge co-inference systems," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.

[41] H. Takahashi, T. Iwata, Y. Yamanaka, M. Yamada, and S. Yagi, "Variational autoencoder with implicit optimal priors," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5066–5073.

[42] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012.

[43] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[44] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[45] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[46] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.